# Graduate Research Plan Statement
# Time-series Interpolation of Scientific Data via Machine Learning

One of the large challenges in High Performance Computing (HPC) is handling data. There is not enough storage capacity to save the data to disk and perform post hoc analysis; the conventional method of analyzing data after a computational simulation has finished. A way to get around this is by doing analysis in situ, that is analyzing and performing visualization tasks while the data is still being generated. Once the analysis is performed then the data is dumped. This can have major drawbacks because scientists cannot reanalyze or come back to parts of simulations after they have thought more deeply about problems and have insights from other work. In essence, scientists have to hope that they got the analysis correct the first time, or they have to go back and re-perform part or all of the simulation again. While in situ analysis has become an increasingly important part of HPC, we have to develop more sophisticated techniques to compute on the exascale.

With the explosion in machine learning research we have seen many sophisticated and complex techniques come out that can greatly help the scientific community. Techniques like super resolution [] can be used as a method to compress images and greatly reduce storage space. Researchers have shown that machine learning algorithms are capable of performing scientific computations []. There has also been work that has used machine learning to help explore the parameter space of scientific simulations []. Techniques like this allow for post hoc exploration that would not be possible without performing more simulations, with those varying parameters.

With this fellowship I plan on developing machine learning algorithms to perform time-series interpolations on scientific data. To achieve this I will break down the project into smaller tasks that will be spread out over the three years of funding. The first part will be to analyze and develop methods for feature reduction so that data can fit onto GPU memory. The second part will be to develop algorithms that demonstrate compression and encoding techniques that further improve on the compression and analyze acceptable limits to this lossy compression. Third, I will explore techniques to perform basic time-series interpolation of scientific data. Fourth, I will research the ability to generalize these techniques and determine the limitations to each, and where they can best be applied. Finally, I will focus on converting these to online algorithms, so that training and fine tuning can be done in situ.

While there has been a lot of work with interpolation [] and frame upscaling [] in the commercial space, the problem is more complex within the scientific regime. Scientific data in HPC is much larger than those conventionally used in machine learning algorithms, with simulations capable of producing petabytes of data a day. Machines do not exist that can store this into GPU memory, where most machine learning algorithms take place. There are many established techniques like t-SNE and Principle Component Analysis (PCA) that are done to reduce feature space for machine learning algorithms. This will be an integral part into determining how to best train data, perform the correct preprocessing, and how to reduce the data space to a size that can fit on modern hardware.

While getting data to fit onto hardware is an important task, we want that data to be as small as possible to reduce training time and reduce the computational load. One of the breakthroughs in modern machine learning is using it for compression. It has been shown that machine learning can be used to perform complex encodings of data that can be used to regenerate the original set. This type of encoding will be important for several reasons. Once we have a good encoding we can reduce our memory burden and free up resources for competing computations. This can also enable more checkpointing for the simulations; where a checkpoint can allow for a simulation to be run from that point instead of starting again from the beginning. With the completion of this milestone scientists will be able to more easily restart simulations and be able to save more data out for post hoc analysis.

Once data has been adequately reduced in size and proper encodings have been developed, focus can shift to developing architectures that will allow for time series interpolation. The goal here is to develop architectures that will be able to predict some of the underlying physics and perform interpolations over large time-steps. Some of the newest machine learning techniques provide methods for this to be possible. Architectures like Recurrent Neural Networks (RNNs), Transformers, and Attention demonstrate the ability for the algorithms to become contextually aware. An important aspect of interpolating scientific data is to understand how parts of the data move within a given system. Conventional methods use momentum of data points to determine how different subsections of data evolve compared to others. These architectures have provided great leaps forward in Natural Language Processing (NLP) with the ability to construct large scale text that is nearly indistinguishable from human writing []. This same contextual awareness can theoretically be applied to the language of scientific analysis. These architectures provide a necessary component that helps enable algorithms to understand the structure of time within scientific computing.

Generalization is one of the most difficult tasks in machine learning. With any algorithmic development it is important to understand its applications and limitations. A deep analysis will need to be performed to determine the extent of generalization and how much fine tuning will need to be done.

Finally, these need to be converted to online algorithms so that fine tuning can happen in situ. This will add flexibility for scientists and decrease the computational demands of the algorithms.

**Intellectual Merit**

**Broader Impacts**

**References**