Reconstructing Plasma Objects: Deep Learning for X-ray Tomography

Jacob Braswell, Isaac Whittaker, Tyki Wada *(BYU)*

Brian Spears, Luc Peterson, Jim Gaffney, Rushil Anirudh *(LLNL)*

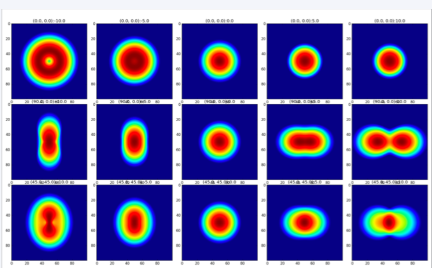
Abstract

In Lawrence Livermore National Laboratory's ongoing efforts to ignite a laser-driven inertial confinement fusion (ICF) target, x-ray cameras are heavily relied on to take a wide collection of images – from different viewpoints, at different energies (or colors), and at multiple times.

Outcomes

Our goal for this project was to develop a deep learning tool to infer the 3D configuration of the glowing plasma from a collection of hyperspectral (multi-color) images at a variety of angles and shapes. This deep learning tool will be utilized towards clean energy research and employed in a Virtual Reality system to encourage local STEM outreach.

A lightweight workflow package to generate ray tracing images was provided beforehand. Though the package would already work as a backend to the VR model, a trained neural network would allow for continuous outputs of images for any given parameter.



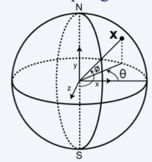
Understanding Plasma

Plasma, the fourth state of matter, is similar to gas but ionized. Most abundantly, plasma is created in the sun as hydrogen gas is fused by pressure, releasing energy and creating helium. To mimic this process on earth, LLNL fires lasers at hydrogen isotopes to cause a confined energy implosion. X-Ray images above are taken at various angles, energies, and times.

Data Collection/Experimental Design

Data was generated using a ray-tracing simulation package offered by LLNL. Shape and Azimuth angles ϕ, θ, and p were used as input. However, optimizing image cost was a challenge towards training our neural network. To be as efficient as possible, specific transformations were made in sampling our data.

ϕ – Uniform
θ – Inverse Cosine
p – Uniform



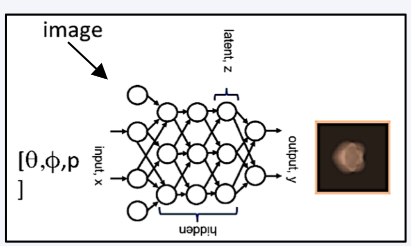
In sampling, both a Latin-Hypercube Design and a Full-Factorial Design were combined. Due to the LHD's joint-independent nature, the latent space was easily sampled using each distribution. Afterwards, any gaps were filled by a thin sample of the FFD.

Data Storage

Images were stored as NumPy arrays using PyTables for fast I/O and ease of use. Storage and computation was completed using the supercomputers on campus (~16 hours). Files were split into two files (train and test); metadata such as parameter name and image shape was also included.

Model Build

After considering several models to generate our images, two models were compared: a Keras sequential model and a Deep Convolutional Generator. Both were evaluated on the basis of loss (MSE), # of parameters, and time to train. All computation was completed on the Google Colaboratory Tesla K80 Cloud GPU.



Model Build Cont..

Both models were trained ~100 epochs with varying batch sizes (KS: 25, DCG: 128). Training was performed by inserting output layers as 25x25 images, then incrementally augmenting to the original 100x100 images. Below is a matrix of loss, parameters to train, and CPU image prediction output time/1000 images.

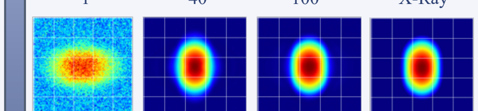
	Keras Sequential	DC Generator
MSE	4.0355e-04	2.3930e-04
Param	13,171,419	12,089,088
Out	8.359 sec	7.689 sec

The DCG model outweighed the performance of the fully-connected network on all fronts. This is likely due to the nature of transposed convolution (an up-sampled gradient calculation is applied to a convolution) to augment an image. Out of sample evaluation performed just as well (loss: 4.0555e-04); the DCG was eventually selected as our highest functioning product.

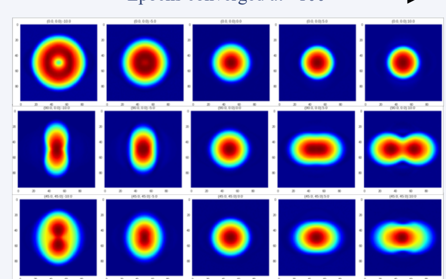
Results

Images of (p: -4.49, θ: 76.29, ϕ: 101.79) at epoch:

1 40 100 X-Ray



Epochs converged at ~100



Conclusion

Preparation: Prior to building the model, careful understanding of plasma angles and storage methods were necessary to efficiently sample the latent space and comply with neural net training.

Model: In comparison to a simple fully-connected network, the Deep Convolutional Generator seemed to maintain the lowest error scores and training time. Additionally, with superlative image clarity and prediction time, the DC Generator seems to be a worthy candidate towards the VR backend system.

Challenges: Understanding the mechanics behind the neural network was the most problematic. Originally, a GAN was proposed due to the nature of the problem. However, it was later recognized that training just a generator on the images without a discriminator sufficed when predicting an image from 3 parameters.

Impact

Fast output: With image output at nearly a hundredth of a second, the loaded model should complement the VR slider amiably.

Continuity: As a result of sampling using a Latin-hypercube design, the model was able to train over a continuous space. This allowed for predicting a continuous X-ray object instantaneously.

Customization: This model was left as a skeleton for future improvement. Parameters such as temperature, energy, etc. may be easily incorporated to affix research value.

Community Outreach: Assisting youth to develop appreciation towards STEM related fields will be the most important impact to deliver. We hope to enable future students to take interest in STEM studies, realize their potential, and add future value to the rapidly expanding world of technology.