Data-driven Design Optimization for Inertial Confinement Fusion Experiments

Women in Data Science

Livermore, CA
March 2, 2020

Kelli Humbird
Luc Peterson, Bogdan Kustowski, Brian Spears
Inertial Confinement Fusion (ICF) compresses deuterium-tritium (DT) fuel to extreme conditions to produce fusion energy

- ICF experiments seek to create immense amounts of energy via fusion reactions
- Reaching fusion “ignition” is challenging; the physics is complex and there are many sources of performance degradation
- We rely on computer models to design experiments, but even our best models are not predictive of all ICF experiments

Fuel capsule converges ~30x

Images: L. Berzak Hopkins
New data analysis tools can improve how we design and understand inertial confinement fusion (ICF) experiments.

Optimize design with simulations

NIF Experiment

Re-optimize in light of experimental evidence
Machine learning can improve the design loop by explicitly updating models using experimental evidence.

- Generate simulation database
- Train ML models to emulate ICF codes
- Calibrate simulation models to experiments
- Search for optimal designs

**Figure:**
- Synthetic Diagnostics
- DJINN Prediction vs. log10 Yield
- Experimental log10 Yield
- Capsule thickness
- Capsule diameter

**Graphs:**
- Comparison of DJINN prediction with experimental log10 Yield.
- Calibration of DJINN models with calibrated and uncalibrated data points.
Machine learning (ML) models are fast representations of expensive simulations

- ML models emulate physics codes by learning the relationship between inputs/outputs
- Models enable us to estimate results where no data exist
- Examples:
  - Polynomial curve fitting
  - Power laws
  - Neural networks (NN)

ML models are fast approximations to expensive simulations
We calibrate ML models using a technique called transfer learning.

Can transfer learning be used to “transfer” between simulations and experiments?
Transfer learning is used to make more predictive models of ICF experiments.

Train NN on large database of cheap simulations

Simulation Inputs

Simulations Outputs

Freeze all but the last layers of the network, retrain on sparse, expensive data

Experiment Inputs

Experiment Outputs
A series of Omega ICF experiments provides a good testbed for transfer learning from simulations to experiments

- The data* includes:
  - 30k low fidelity 1D LILAC simulations
    - Spans a 9D input space with varying laser pulse & capsule dimensions
  - 23 High fidelity simulations
  - 23 experiments with measurements of yield, bang time, Tion, rhoR, burnwidth

*Data provided by Varchas Gopalaswamy & Riccardo Betti
NN+TL: predict high-fidelity simulations with low computational cost

- Low-Fi. NN
- High-Fi. NN (train)
- High-Fi. NN (test)

Graphs show:
- High-fidelity Yield
- High-fidelity BangTime
- High-fidelity BW
- High-fidelity Tion
- High-fidelity RhoR

Legend:
- Low-Fi. Predict
- High-Fi. Train
- High-Fi. Predict
NN+TL: more predictive of future Omega experiments than simulations

Experiment DJINN models can predict future Omega experiments
Each model suggests a different optimal* implosion

DJINN: Optimal Laser Pulses

Optimal Capsules

- Low-fidelity NN
- High-fidelity NN
- Experiment NN

*Maximum Yield \cdot (\rho R)^2
Each model suggests a different optimal* implosion

Max 1D Yield
Fix LPI but maintain velocity
Control hydro. instabilities at higher $\rho R$

DJINN: Optimal Laser Pulses

Optimal Capsules
- Low-fidelity NN
- High-fidelity NN
- Experiment NN

*Maximum Yield \cdot (\rho R)^2
We are using machine learning to create more predictive models by integrating ICF simulations and experimental data

- ML models can be queried millions of times to rapidly search for simulations to optimize designs

- Transfer learning is a novel method for creating predictive models

- You can try these techniques on your own data:
  - Download our neural network software at github.com/llnl/djinn
Transfer learning enables us to continuously improve our predictive capabilities by updating our models with data.

**Simulations won’t change** their map, although input transformation models (like multipliers) can rotate and scale it.

**Machine learning methods can update** the map with observations not in the original simulation model.
Applying transfer learning to NIF experiments is more challenging

- For Omega direct drive experiments, capsule-only simulations are analogous to the experiment

- For NIF indirect drive experiments, hohlraum simulations are analogous to the experiment
  - TL with hohlraums is challenging for several reasons:
    - Hohlraum simulations are very expensive (can run ~5k in a month timeframe)
    - The NIF experimental design space is extremely large (meaning lots of inputs, and therefore lots of simulations needed for the NN)

- So how can we use TL to help improve predictions of NIF experiments?
We can explore a different approach to transfer learning from simulations to experiments using autoencoders

- Autoencoders are a type of neural network traditionally used for data compression
  - Remove redundant information and learn correlations between observables

- Autoencoders are simply specialized mappings from outputs to outputs

![Diagram of autoencoder process](image)

- Compress: Observables → Compressed Data = “Latent Space”
- Decompress: Compressed Data → Observables
Can we transfer learn an autoencoder to map from *simulation outputs* to *experimental outputs*?

Step 1: Train an autoencoder to map from simulation outputs to simulation outputs with large database.
Can we transfer learn an autoencoder to map from *simulation outputs* to *experimental outputs*?

Step 2: Use pre-shot simulation outputs and actual experimental outputs to transfer learn the autoencoder.
The autoencoder gives a “correction” to simulation predictions such that they are more consistent with previous experiments.

Simulation inputs $\rightarrow$ Simulation Outputs $\rightarrow$ Expected experiment outputs

Step 3: Use our “calibrated” predictions to search for optimal experimental designs.
We can optimize experiments using a map that is updated as we gain experimental data.

1. **Sim. inputs**
2. **Run simulation**
3. **Simulation outputs**
4. Use optimization algorithm select new inputs
5. **Expected exp. outputs**
6. Run optimal experiment
7. Update AE
A thought experiment: Use transfer learning to guide a NIF campaign to maximize ITFX\~Yield*DSR^2

Phase 1: Explore and Build Model

- “Experiment”: a Fake Model of NIF:
  \[ \text{Exp}(x) = \text{Hydra}(Bx) \]
  
  \( B \) is a random matrix

Phase 2: Exploit and Optimize*

- Update Model with Data
  - Transfer learn the autoencoder

- Pick the Next Experiment
  - Search for max Yield*DSR^2

DSR=down scatter ratio

*Similar to Google Optometrist Algorithm
The transfer learned autoencoder gets better at predicting the experiments as it acquires data.
Transfer learned model finds the optimal design within ~6 experiments

- We repeat the design optimization loop for several random realizations of the “experiment” ground truth:

\[ \text{Exp}(x) = \text{Hydra}(Bx) \]

- On average, the TL optimization loop enables you to find the true optimal with ~6 experiments; the simulation only model gets within 5% of the true optimal
Preliminary results indicate autoencoder TL is possible with the existing NIF database

- For the Bigfoot campaign we have a set of simulation predictions and experimental observations for 13 experiments

- An autoencoder trained with only 6 observables transfer learns to the experimental data accurately

- Next step: use the autoencoder mapping to correct predictions for the next Bigfoot shot, and see if it is more accurate than the simulation-only prediction
New data analysis tools can improve how we design and understand inertial confinement fusion (ICF) experiments.

Optimize design with simulations

NIF Experiment

Re-optimize in light of experimental evidence
We design experiments to achieve a specific goal, such as increased areal density or high yield

- In ICF, we use simulations to search for experimental inputs (laser pulse, capsule) that achieve our goal

- This can be expensive and challenging as the NIF design space is large, and simulations aren’t accurate everywhere

- Machine learning can help us find optimal designs faster
New data analysis tools can improve how we design and understand ICF implosions

Optimize design with simulations

NIF Experiment

How can we better explore vast design spaces for “optimal” implosions?

Are these consistent?

Are there other explanations for the data?

How do we use experimental data to update our models?

Re-optimize in light of experimental evidence
The transfer learned autoencoder gets more accurate as experimental data is accumulated.

- 5 exp
- 10 exp
- 15 exp

**Sim. Only TL AE**
Alternatively, we can learn the mapping from simulation observables to experimental observables via an autoencoder.