SAND2018-13171PE

Predicting Fragment Aerodynamic Drag with Deep Learning

PRESENTED BY

Peter D. Yeh

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Understanding range of fragment flight

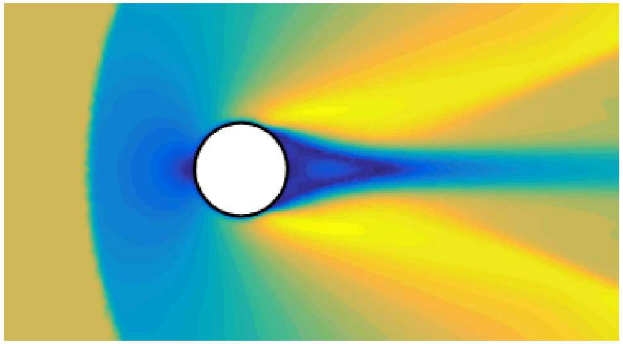
Explosive fragments fly at supersonic speeds

Current methods assume single drag coefficient

Geometry is complicated, high aspect ratio

Fragment-air interaction leads to tumbling and chaotic motion

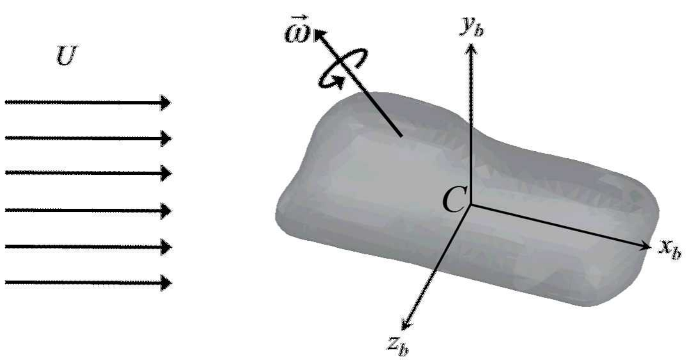
Goal is to characterize range of a set of flying fragments

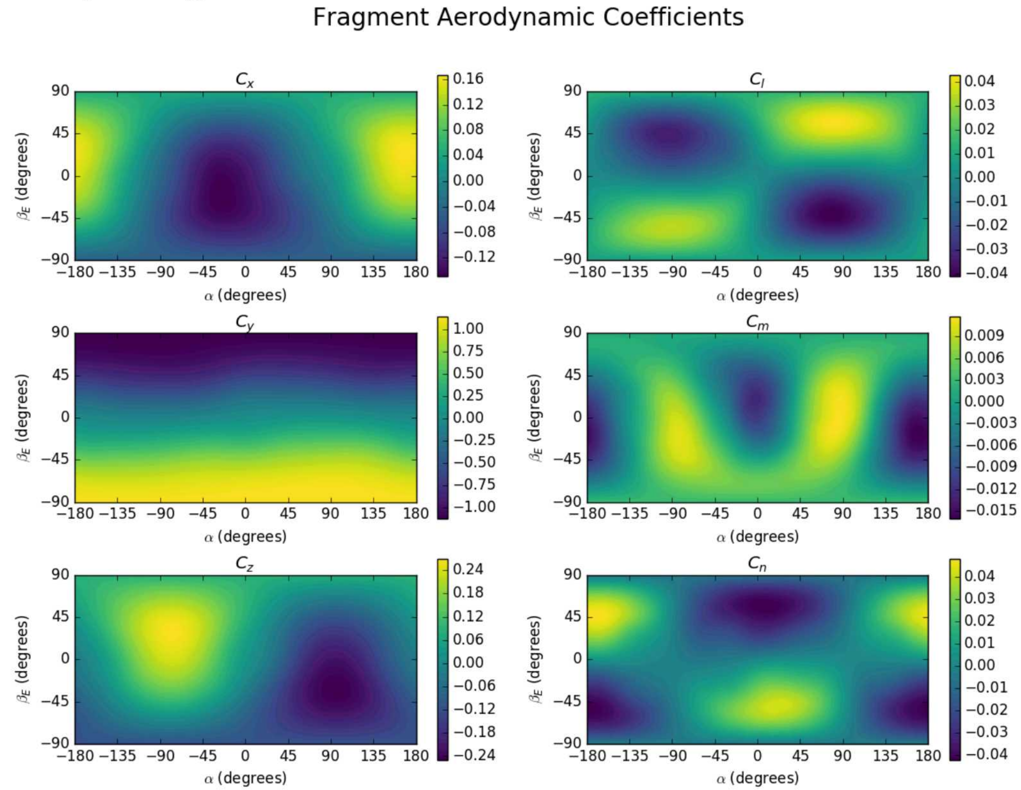


Previously developed fragment flight simulation procedure

1) Compute aerodynamic coefficients at all orientations with high fidelity solver.

2) Compute trajectories with rigid body integration.





An explosive may generate over 10,000 fragments. Simulating all of them is prohibitively expensive!

How can we speed up aerodynamic calculation? Deep Learning!

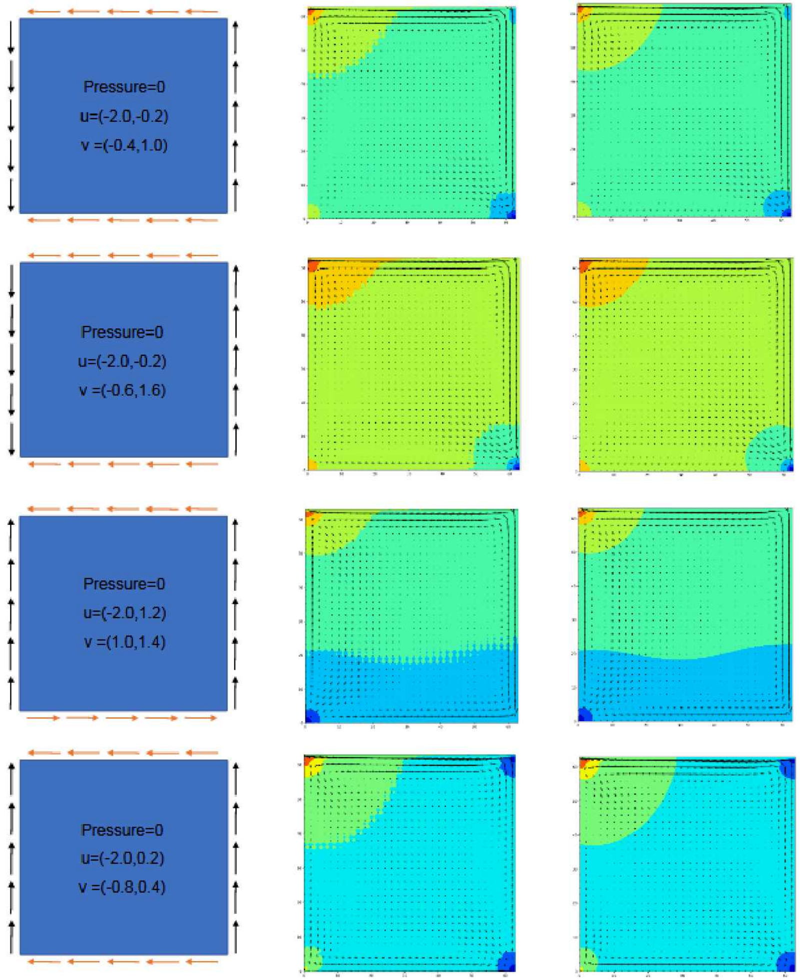
Lid driven cavity solution approximated using Deep Learning (Stanford, 2017)

Used Generative Adversarial Networks (GANs), adapted pix2pix algorithm

Achieved **orders of magnitude** speed-up in inference time

Approximating aerodynamics using deep learning shows potential

Boundary Condition	Deep Learning	CFD Simulation



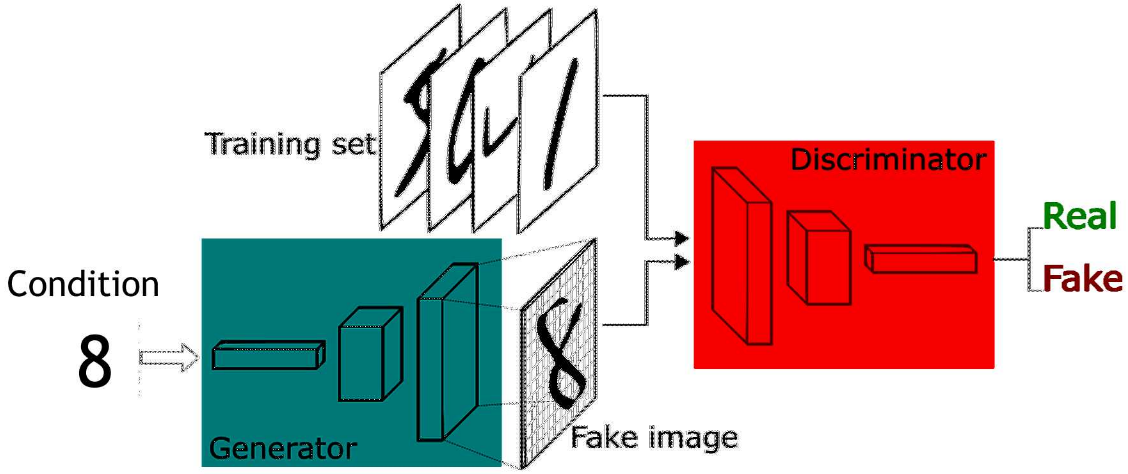
Farimani, Gomes, Pande 2017

Generative Adversarial Networks: a game theoretic approach to machine learning

Generative Adversarial Networks (GANs) pit two competing neural networks against each other

- **The generator**, tries to mimic real results
- **The discriminator**, tries to identify mimicked results from real results

Loss function dictates solution convergence



https://skymind.ai/wiki/generative-adversarial-network-gan

Two networks enter, One network leaves

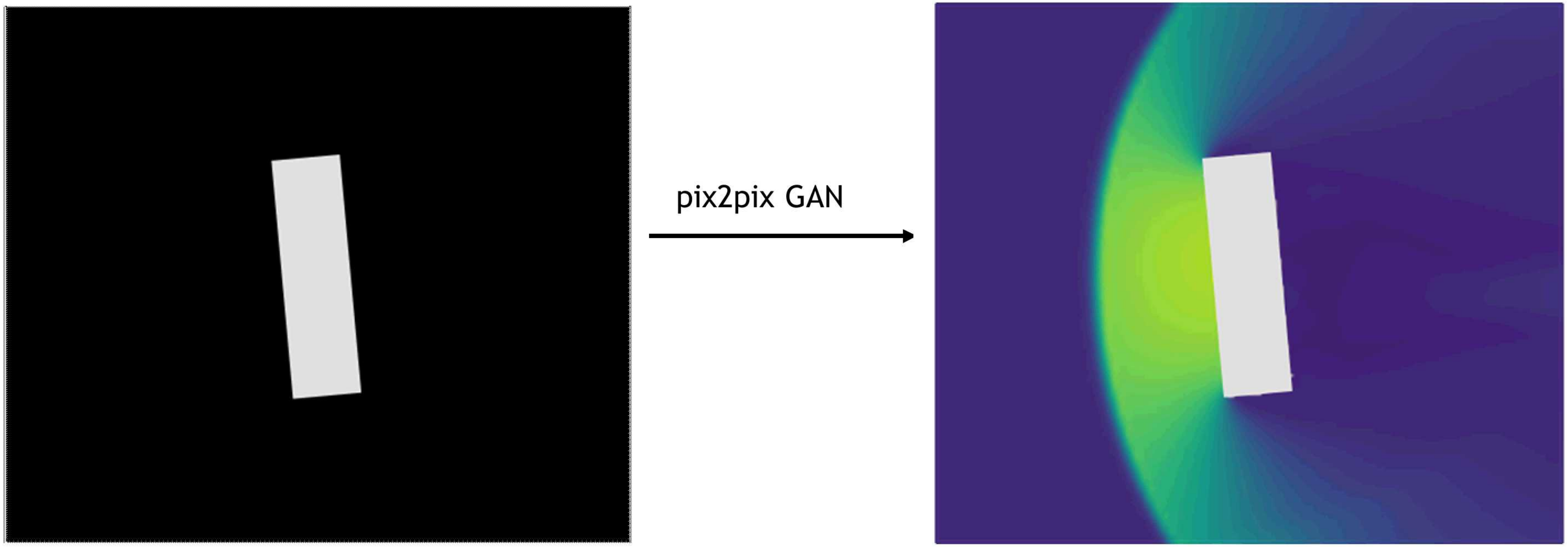
Generative Adversarial Networks (GANs) learn to mimic complex systems with wide applicability

High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs (2018)

Our first attempt at using a GAN for flow prediction

Train pix2pix GAN using computed flow solutions as ground truth

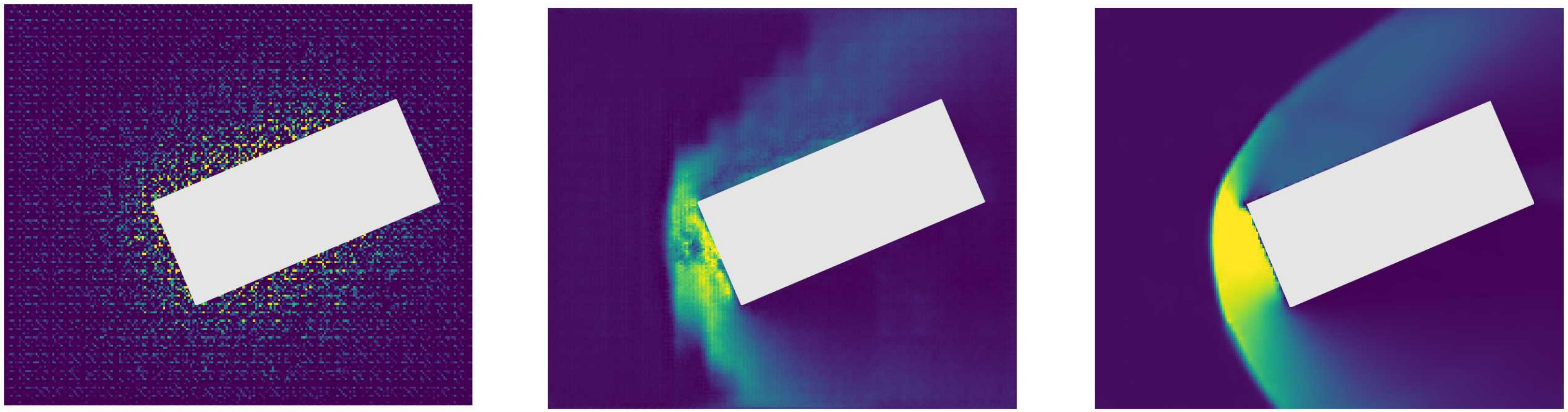
- Simulated 1000 rectangles with random orientations & aspect ratios, Mach 5 external flow
- 900 training examples, 100 held out test examples
- Pressure fields calculated with compressible Euler equation solver (CE Solver)
- Ideal gas assumption for simplicity



Input Data

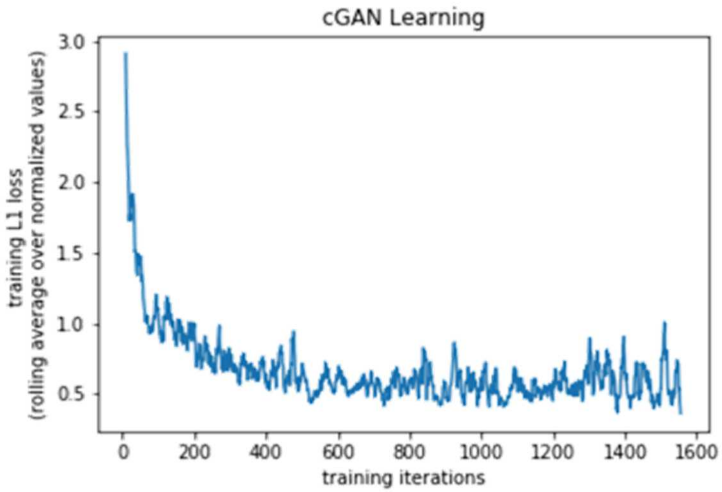
Intended Outcome

GAN-generated solutions look promising



Training Iteration 1

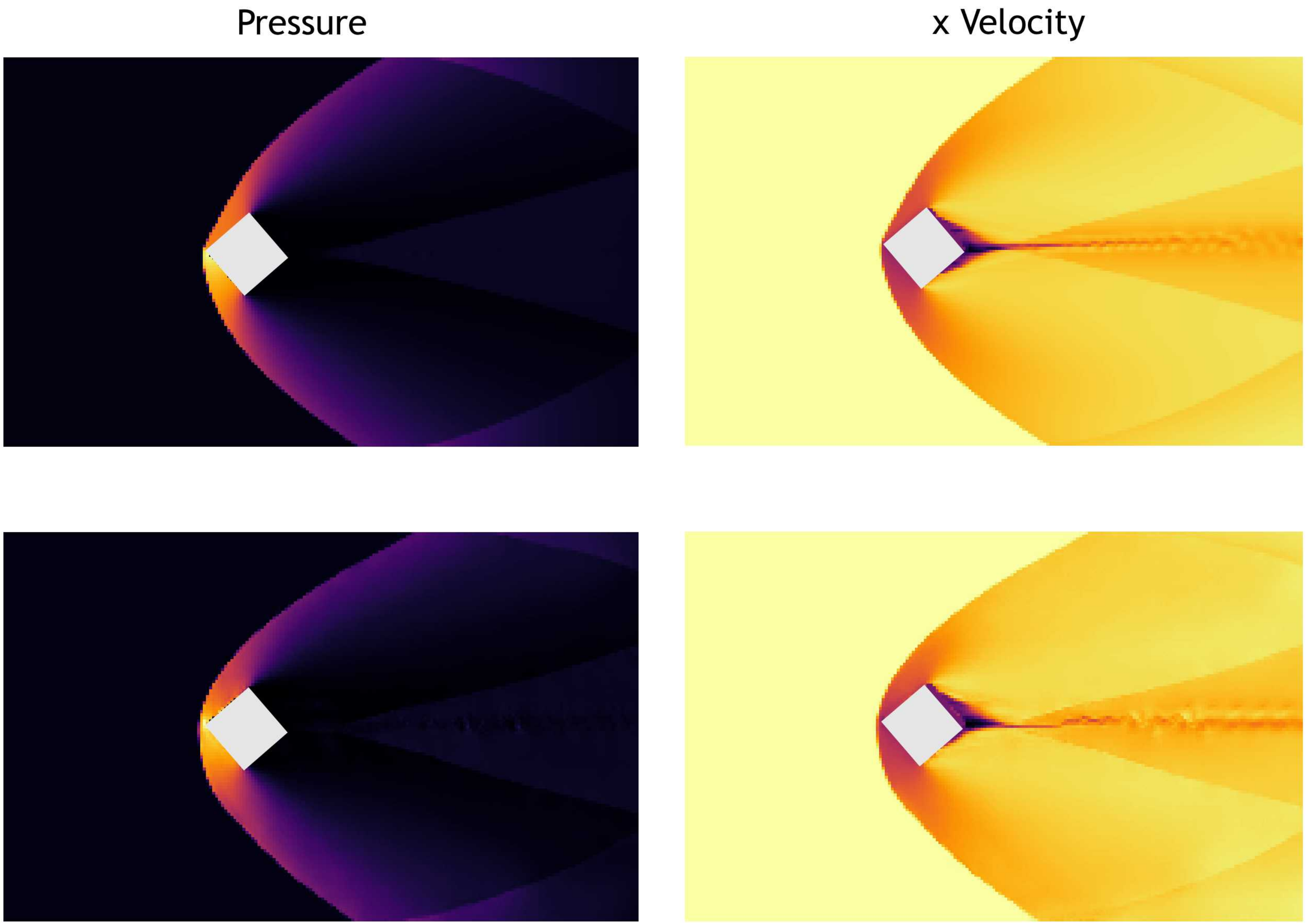
Training Iteration 200



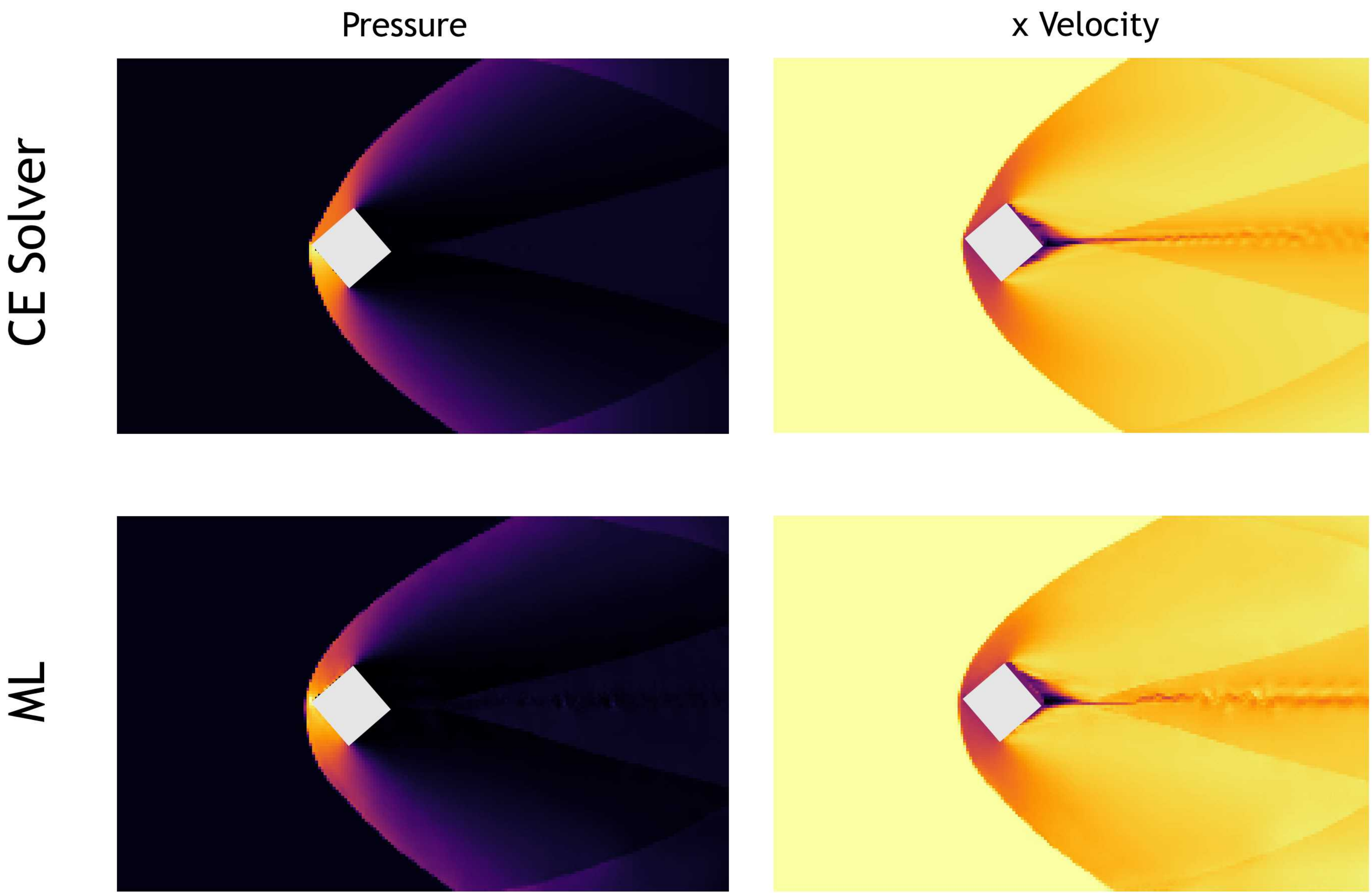
Absolute error from CE Solver decreases over training time

Generator improves over time eventually producing visually accurate pressure maps

Can you tell which flow field is generated by a GAN?



Can you tell which flow field is generated by a GAN?



GAN successfully generates visually accurate flow approximations

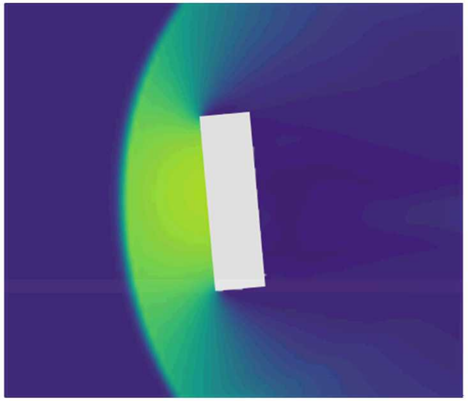
Predicted pressure is close on leading edge, but drag is inaccurate

GAN successfully approximates the pressure map along the leading edge (left) of the object

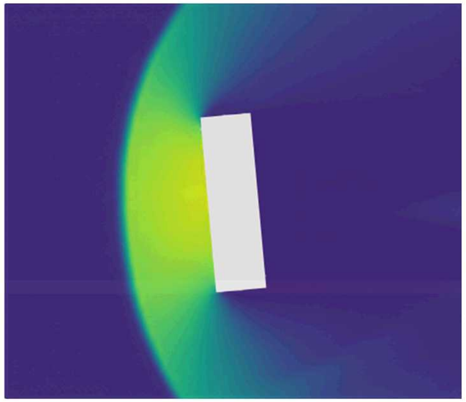
Larger error behind the object due to unsteady wake and fluctuating lower pressure

Inaccurate drag calculation despite close values in pressure

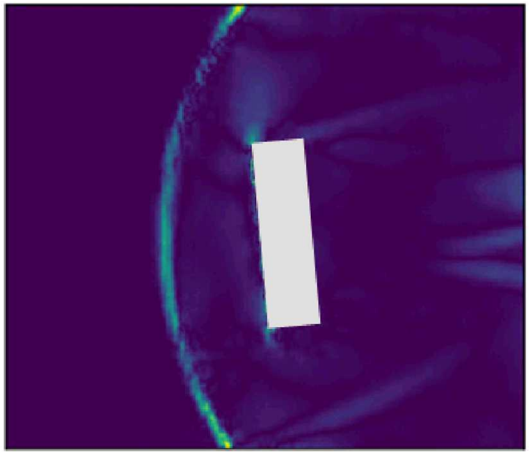
CE Solver



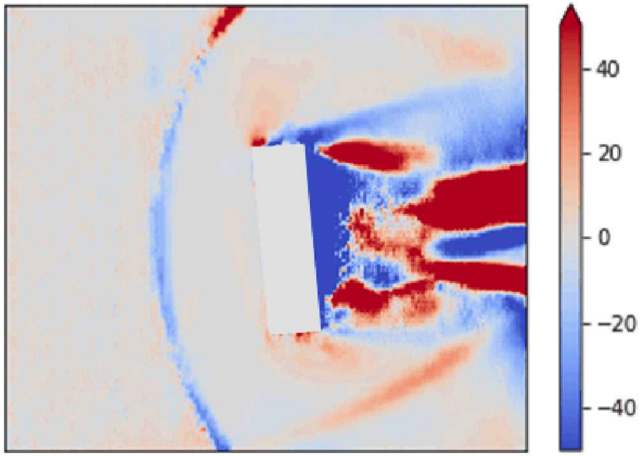
ML



Pressure Abs. Difference

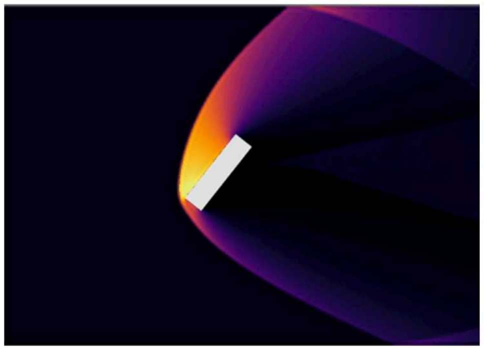


Pressure Percent Difference

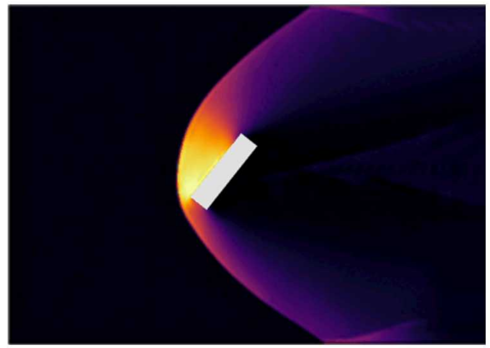


Improvement 1: Add physics loss term to loss function

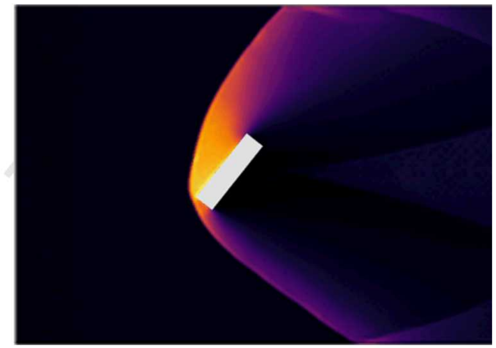
CE Solver



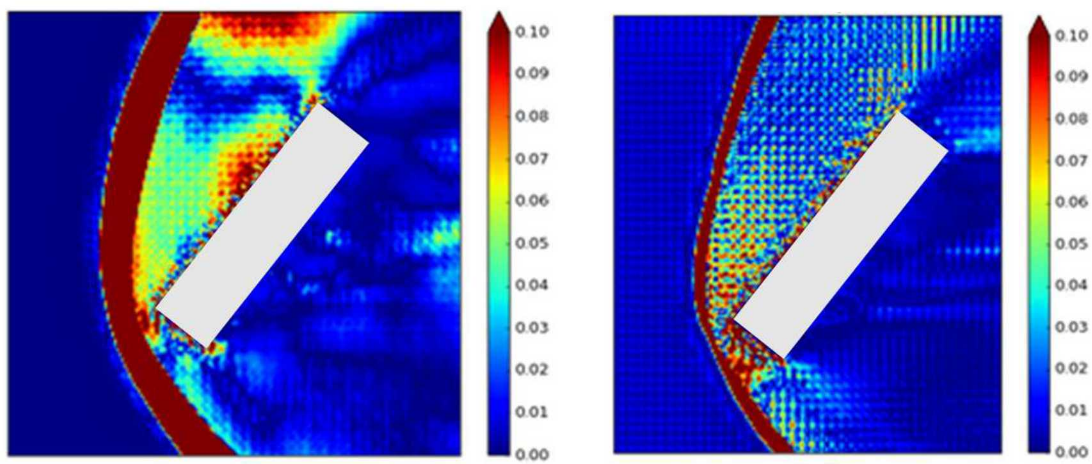
GAN Loss



GAN Loss + Physics Loss



Error



Enforce momentum and mass conservation in generator

Punishes model for violating physics constraints

Improved accuracy in pressure field prediction

Improvement 2: Include force loss term in loss function

Compute Drag, Lift, & Torque

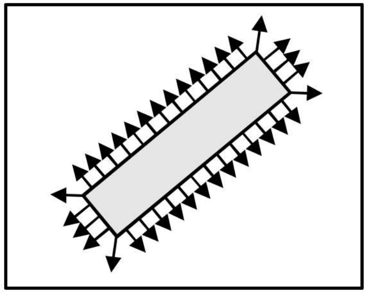
- We take the gradient of the object to get the surface normal
- Using predicted pressure field and the surface normal, we calculate the forces on fragment
- GPUs make this a very fast calculation

Penalize model if computed drag, lift, and torque differs from CE Solution

Total loss function is a sum of GAN loss, physics loss, and force loss

- Physics-informed model

$$drag = \Sigma P n_x$$
$$lift = \Sigma P n_y$$
$$torque = \Sigma P \boldsymbol{r} \times \boldsymbol{n}$$



Normal to object

Successful Deep Learning prediction of fragment aerodynamic forces

Total Loss = **GAN Loss** + weights * **Physics Loss** + weights * **Force Loss**

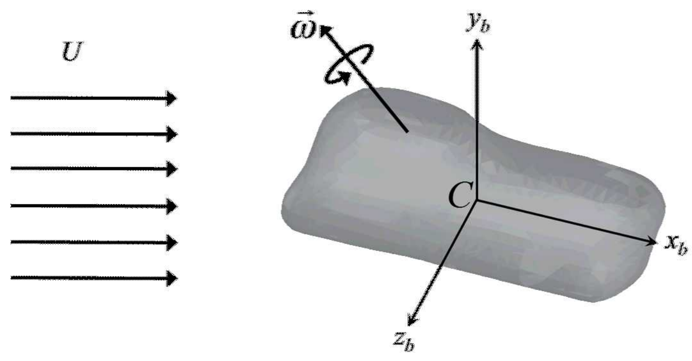
Appropriate adjustment of weights leads to successful predictive model

	Mean Relative Error vs CE Solver
Drag	1.87%
Lift	5.63%
Torque	2.29%

ML predictions approximate CE solver results within 6%

Conclusion and Future Work

Original goal: fast approximation of fragment drag

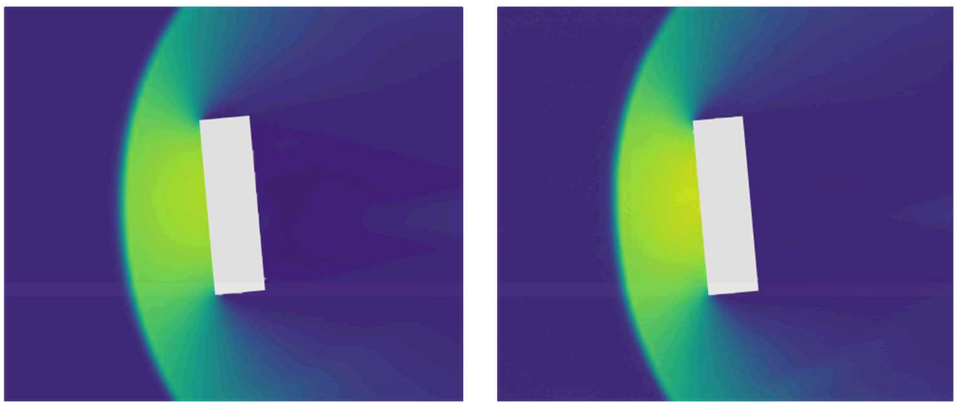


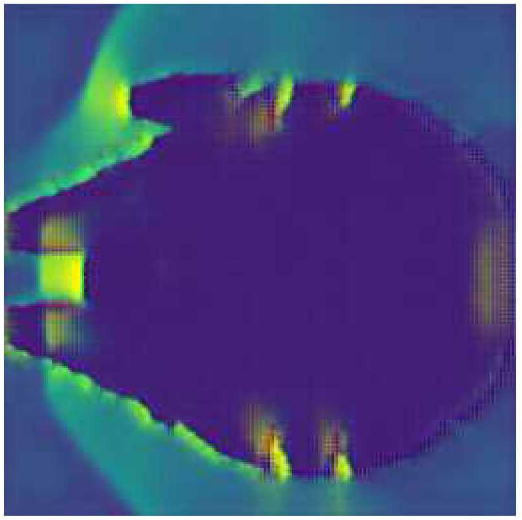
Demonstrated a strategy for deep learning aerodynamic drag

- 2D rectangles
- Physics-informed model
- Generalization and extension to 3D in progress

Limitation: ML model is only as good as its training data

CE Solver ML





Acknowledgments

Sandia Interdisciplinary Machine Learning Research Team (SIMLR)

- Kevin Potter
- Cari Martinez
- Emily Donahue
- Matthew Smith
- Charlie Snider

Engineering Sciences

- John Korbin
- Steve Attaway