Learning physics in the new age of ML

Jay Pathak
Director, Research & Development, Office of CTO
November 2021

Ansys

Agenda

- History of Machine learning
- Machine learning in new age: 2015-Present
- Evolving trends in deep learning
- Challenges of learning simulation
- Ansys Perspective: Key takeaways
- Research Projects

History of Machine Learning

Before 1950s
- **Statistical methods are discovered and refined**
- Bayes Theorem
- Least square

1950s
- **Machine learning defined by Arthur Samuel**
- Games such as Checker and Tic & Tak

1960s
- **Bayesian methods & probabilistic inference in ML**
- Limitations of neural networks

1970s
- **AI Winter**
- Automatic differentiation (AD) of discrete connected networks

1980s
- **Knowledge driven machine learning**
- Geff Hinton's rediscovery of back prop
- RNN & RL

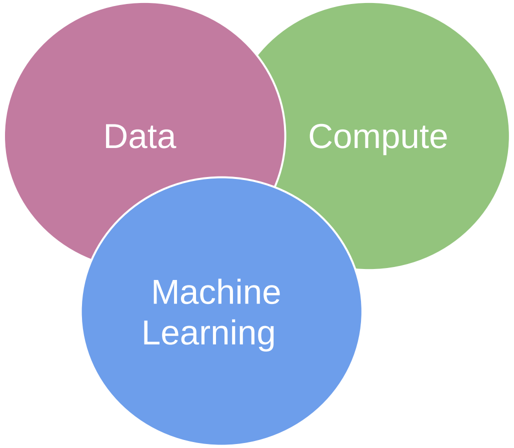
1990s
- **Data driven machine learning**
- Support Vector machine & Random forest
- IBM Deep Blue beats world champion
- RNN and LSTM become popular

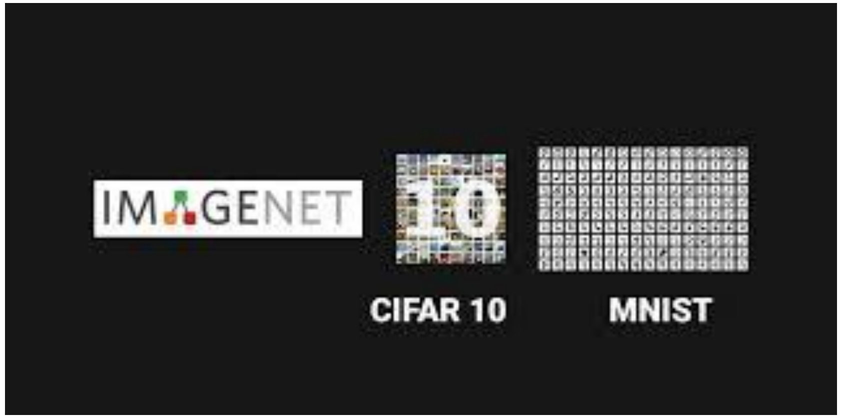
2000s
- **Kernel methods & Unsupervised learning**
- Torch library is born
- ImageNet conceived accelerates modern ML and Computer Vision

2010s
- **Deep Learning becomes widespread**
- Kaggle competition
- Google Brain identifies animals in YouTube
- DeepFace from Facebook
- AlphaGO beats best Go player
- NLP takes a stride

2010-Present: Machine learning

- Data storage is cheap
- Almost infinite compute capacity with race in GPUs/TPUs





Open source datasets which accelerated the pace of research

- Open source and ML Frameworks and datasets
- Online paper repos through Arxiv
- Growing community through top AI/ML conferences

NIPS

ICML
International Conference
On Machine Learning

ICLR

Top ML conferences

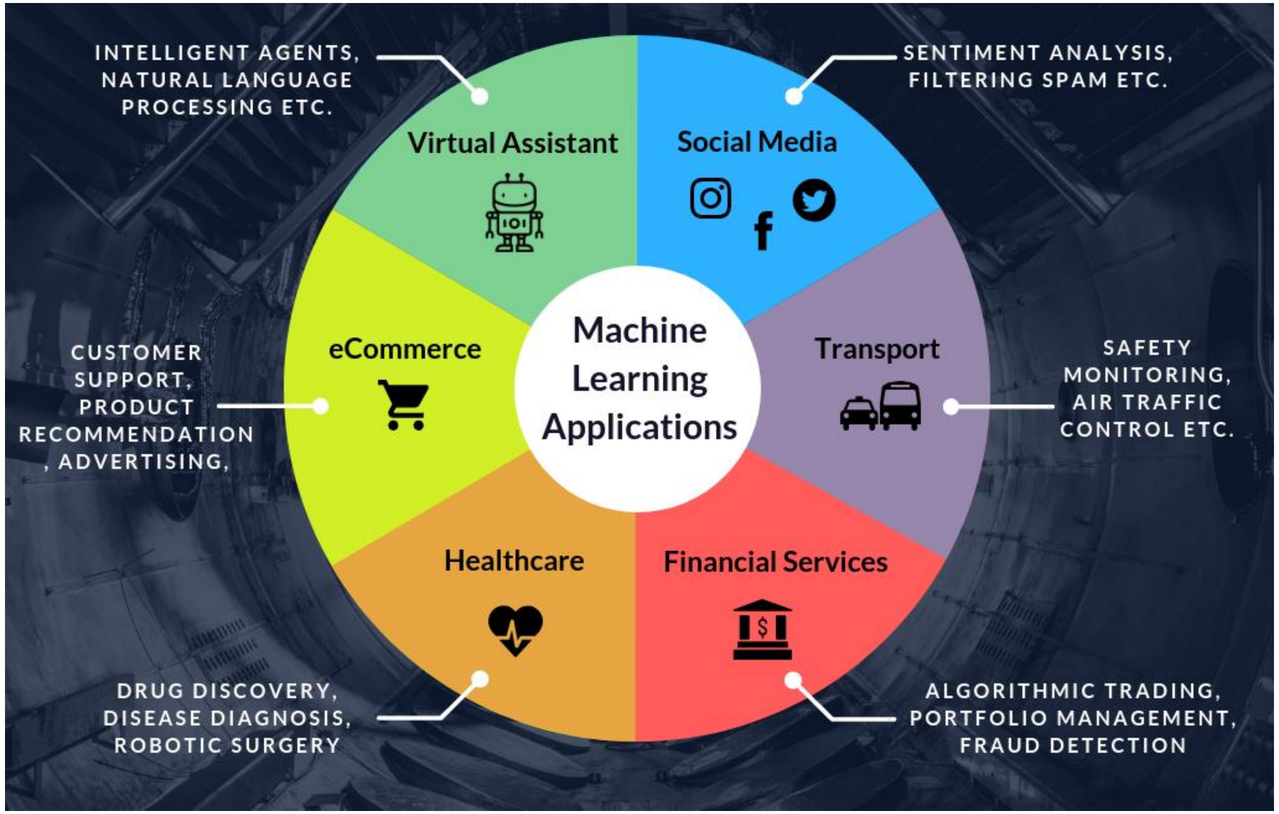
TensorFlow

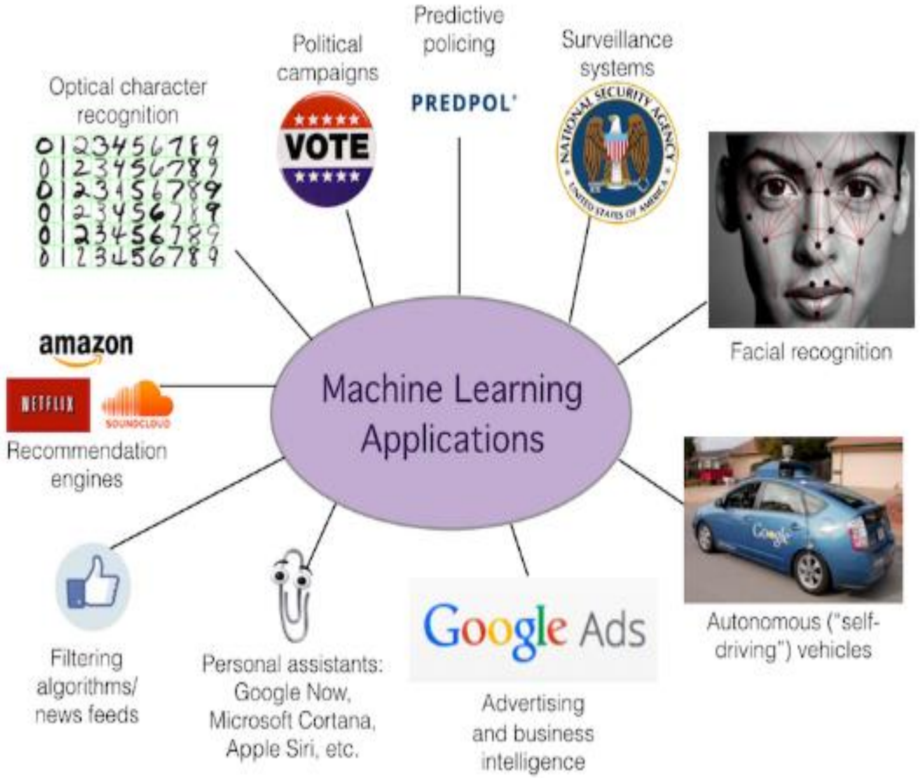
PyTorch

Open source ML frameworks that survived test of time

Ansys

2010-Present: Successful ML applications





2010-Present: Notable failures of ML

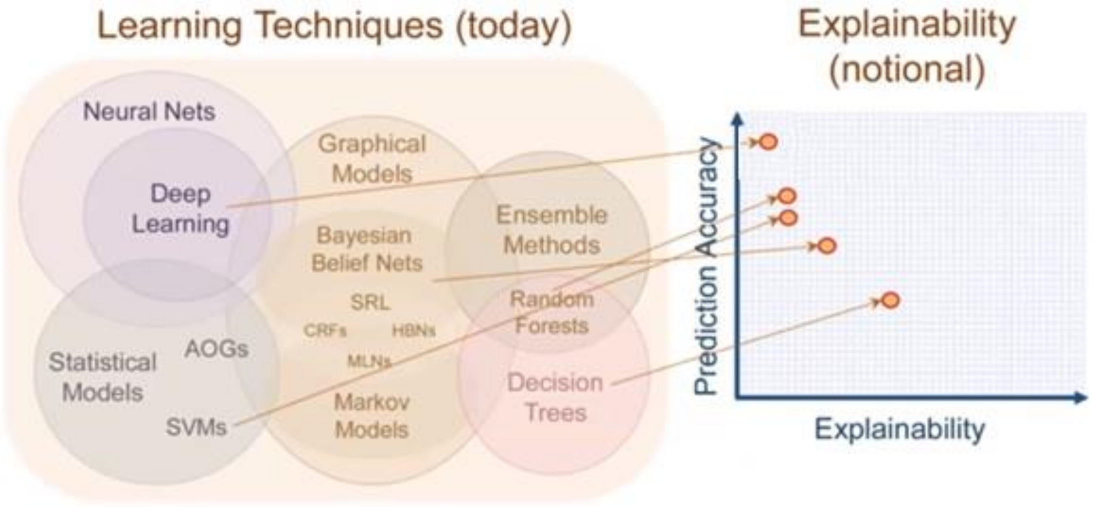
AI/ML is not living up to its hype

- Does deep learning always have overfitting/ generalization problem?

Machine Learning in new age: 2015 - Present

- Towards accuracy, explainability and generalizability
- Big data -> Useful data
- Evolving trends
  - **Data efficient learning**
  - Self-supervised learning
  - **Unsupervised learning**
    - Multi-task learning
    - Multi-modal learning
  - Meta learning
  - From induction to deduction
  - **Imposing hard constraints**
  - **Combining Bayesian methods into deep learning**
  - Learning without forgetting & building a knowledge graph

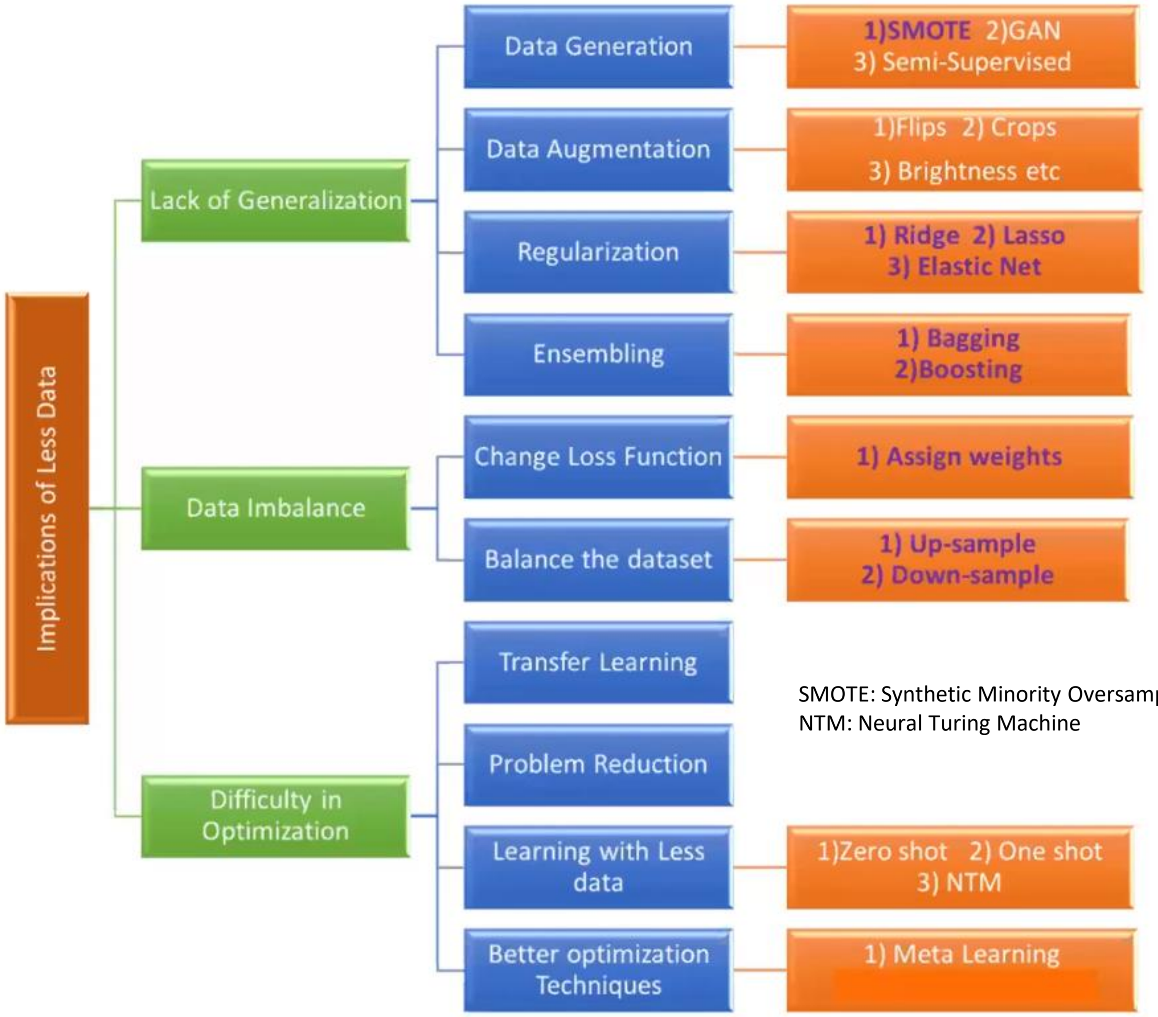
Accuracy *vs.* Explainability



[https://www.researchgate.net/publication/329558375_An_AI_degree_with_an_OpenCourseWare_first_draft]

Data efficient learning

- Lack of data causes 3 problem?
  - Generalization
  - Imbalance
  - Optimization
- Ways to deal with them
  - Read the article

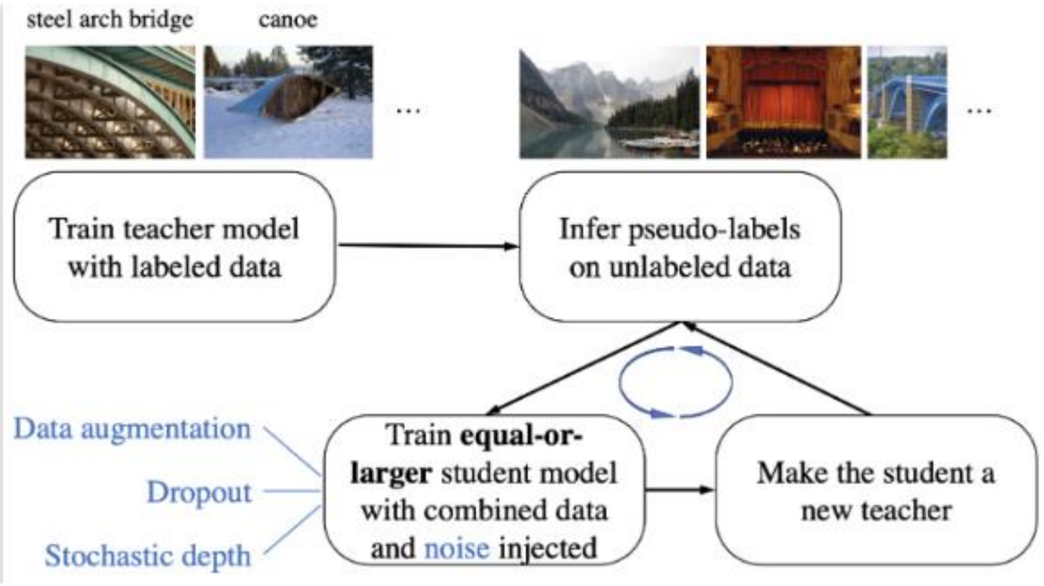


SMOTE: Synthetic Minority Oversampling Technique
NTM: Neural Turing Machine

https://towardsdatascience.com/breaking-the-curse-of-small-data-sets-in-machine-learning-part-2-894aa45277f4

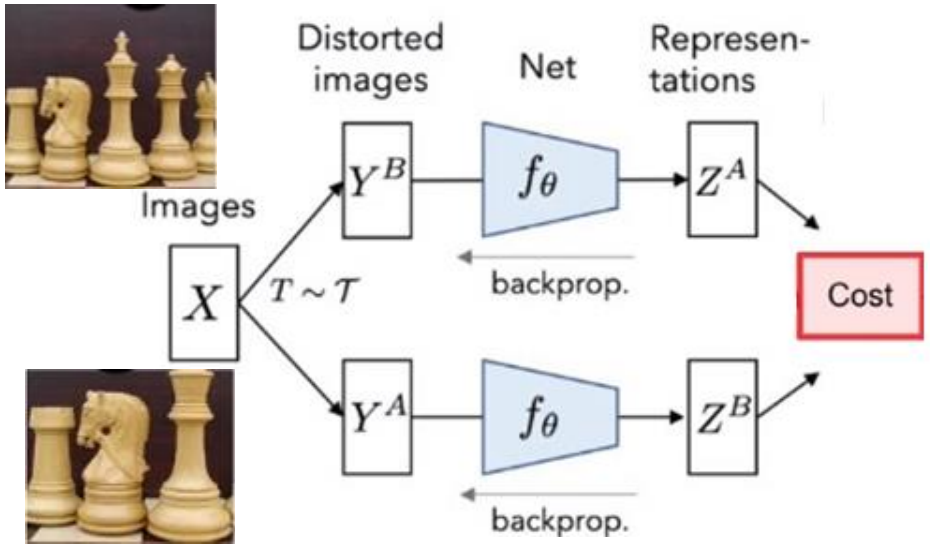
Data efficient learning..

- Semi-supervised learning
  - Self training with noisy student
  - Improves robustness and accuracy of ImageNet state of the art model
  - Teacher starts off overfit turns gradually into robust general-purpose model



https://arxiv.org/abs/1911.04252

- Self-supervised learning
  - Future of representation learning
  - Data augmentation: crop, rotate, noise
  - Learn similarity in z space: Minimize redundancy between embedding variables and maximize info content between z vectors



LeCun: https://arxiv.org/abs/2103.03230

Realities of Simulation

- Numerical methods have been there for last 50 yrs
  - Approximations are there but trust is well established
- Speed is desirable but accuracy is indispensable
- Physics needs hard constraints to be satisfied
  - Everywhere in the domain, say observing a random point, plane or a region of interest
- Typical simulation models
  - 100-1000 parts
  - 1000 connections
  - One or more physics with a couple of idealizations/physics models
- Commercial simulation software such as ANSYS is not only solving PDEs but
  - Constraint equations
  - Heuristics
  - Physics models
  - Couplings across physics
  - Many more..

AI/ML Challenges applied to simulation

Data Hungry

Simplistic

Stochastic

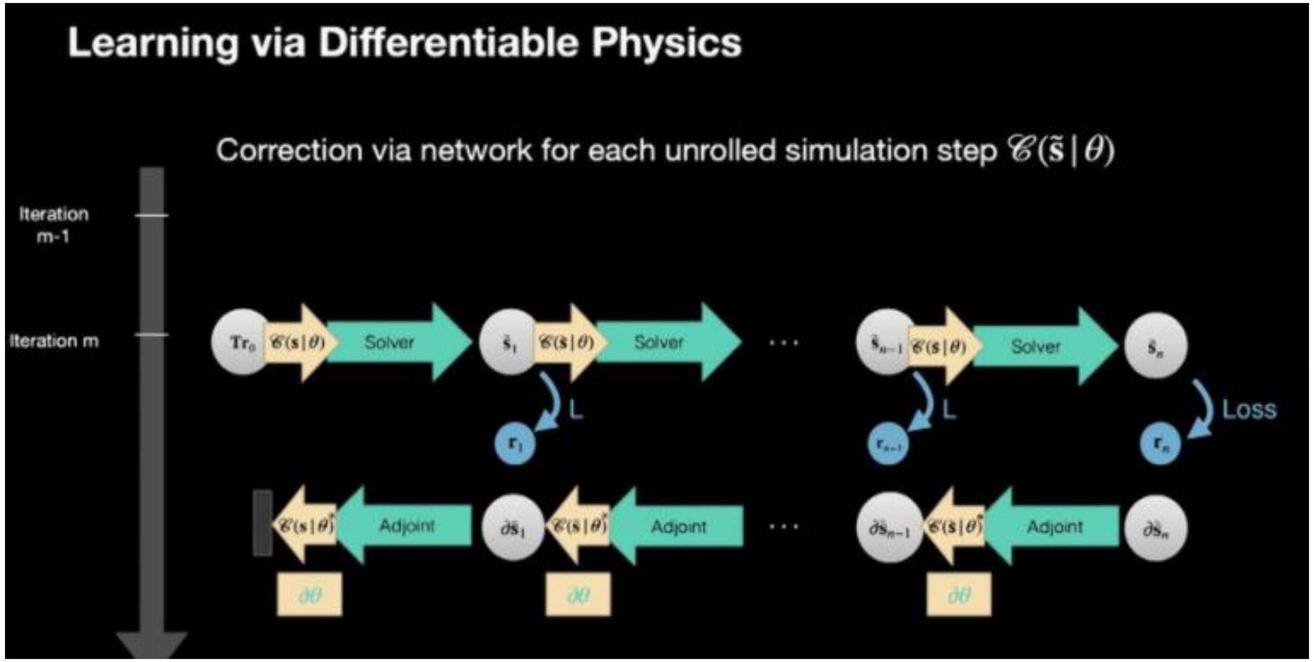
- Scaling data driven methods to different orders of geometric & physics complexities
  - How to move from specific application to strong generalizations?
- Accuracy is key, don’t want to sacrifice or trade with speed
- Even with zillions of simulation and access to data, problems can be unique?

Ansys Perspective: Key Takeaways

- We cannot replace PDE solvers in near term
  - An initial cheap solution is critical to fix the intractable input space
  - Large nonlinearities always a learning challenge, use the solvers first
  - Map coarse solutions to high fidelity solutions using ML
    - Several reputed papers on this: Google, Tech Univ. Munich, Caltech, Princeton
- Better software engineering & owning data distribution problems
  - Scale by domain decomposition and focus on building blocks
  - Reduce redundancies in datasets and data augmentation to create quality
  - Smaller networks, focus on scalable training & accuracy/Num of parameters
- Local learning is key for strong generalization, and it solves data problem too
  - One shot, zero shot learning with few simulations
- Learning with differentiable geometry, physics, rendering etc.
  - Use it with caution!
  - Very nonlinear physics such as rigid body dynamics, explicit simulations are challenges for ML
  - Feature extractions to handle nonlinearities: Koopman Operators, Fourier neural operators etc.

Ansys Perspective: Key Takeaways..

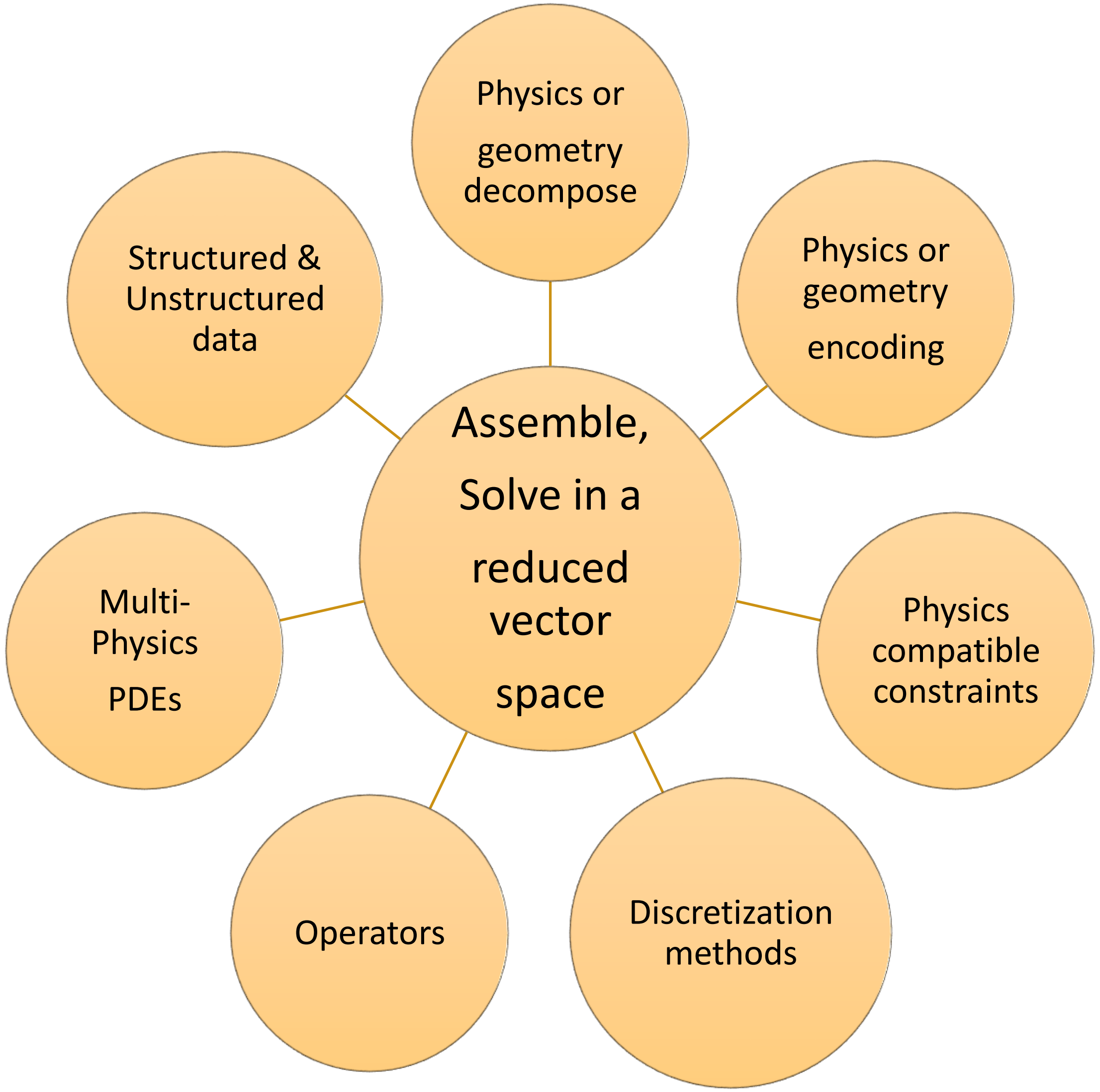
- Imposing hard constraints into the learning process always helps
  - Pinn family of Networks: Physics based multi-objective losses are great!
    - Original Pinn Papers
    - Accelerating learning with data losses followed by physics-based losses for fine tuning: DeepONet
    - Nvidia SimNet
  - Surrogate networks
    - Solver in loop: Predictor-corrector networks
  - Output layer-wise outputs in networks
    - Ansys topo project: Volume constraint
  - Implicit constraints in networks: Deep equilibrium models
    - Zico Kolter etel: https://arxiv.org/abs/1909.01377
    - Equality and in-equality constraints into networks: https://arxiv.org/abs/2104.12225
- Optimizing/Solving during test time
  - Solve/optimize after initialization from actual solver
  - Don't solver from scratch: Ansys ML based PDE Solver



Nils, Tech Univ of Munich, Neurips 2020: https://arxiv.org/abs/2007.00016

Key Takeaways: Ansys ML Perspective..

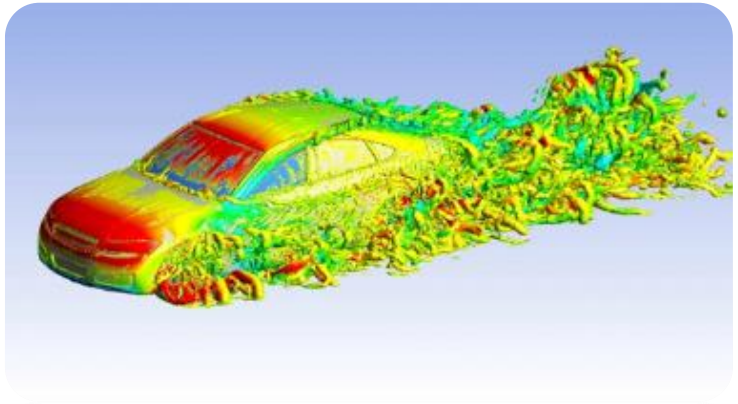
- Army of learned networks
- A controller/assembler to bring on-board one or many learning models based on context
- Must solve again to make sure problem specific aspects covered during inferencing
- **Not a typical inference, optimization during inference time**



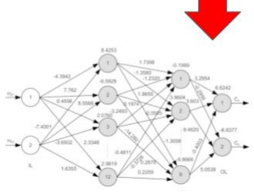
Research Projects

Faster 3d Turbulence

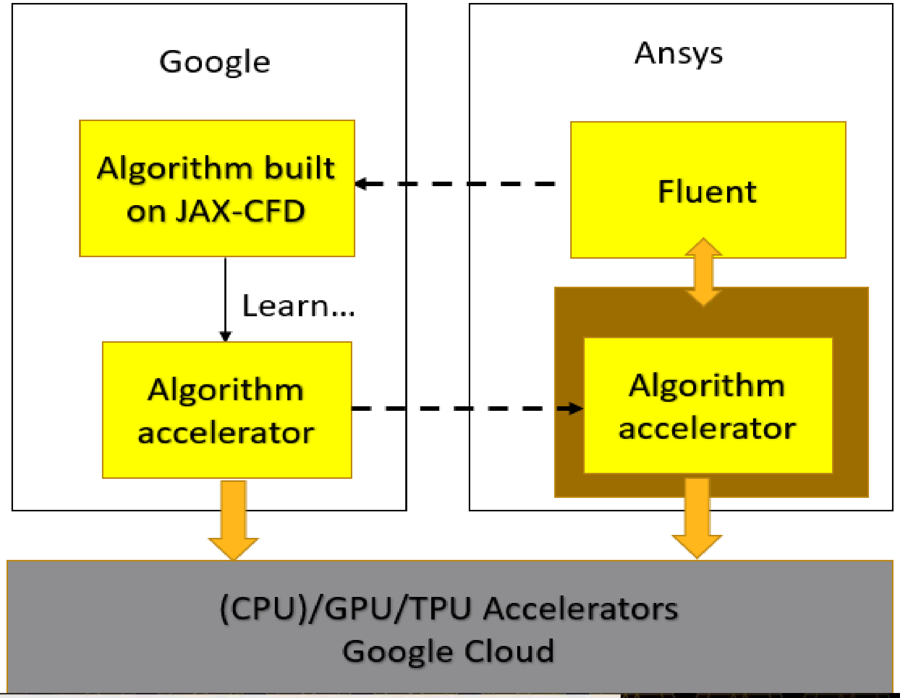
- Goals: Accelerated turbulent flow computation (10x-100x) in Fluent for industrially-relevant scenarios
  - Google to build an accelerator for Fluent where high-fidelity solutions can be solved on a coarse mesh
  - Google published Pnas Paper: Learning interpolation from high fidelity low fidelity for 2D turbulence
  - Use Fluent UDFs
- Interpolation based scheme simply does not scale to 3D
  - Filters increase(2n^2.K^n) from 128 to 1152!
- Looking at learning corrections using a predictor-corrector network(Nil’s 2020 Neurips)

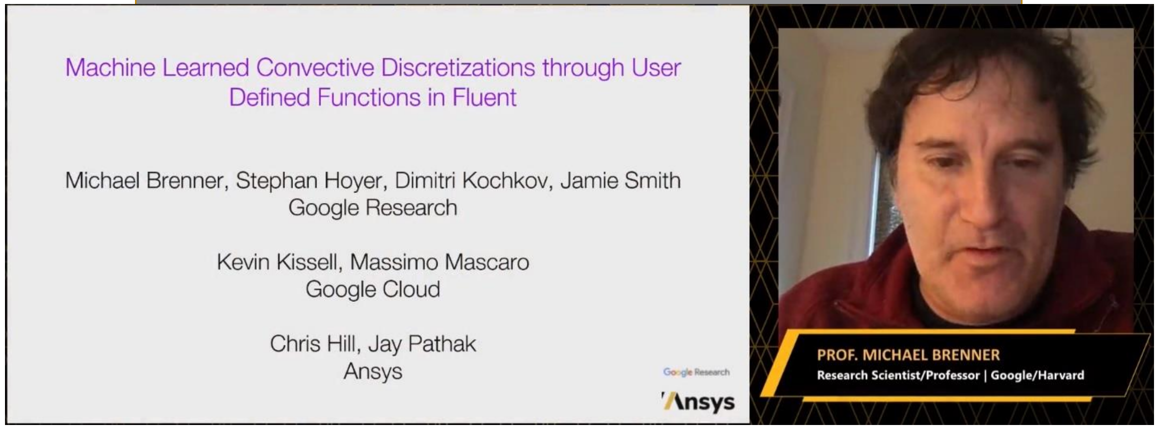


$$\frac{\partial^2 v}{\partial x^2} \approx \frac{v_{i+1} - 2v_i + v_{i-1}}{\Delta x^2}$$



$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{j} = \sigma$$

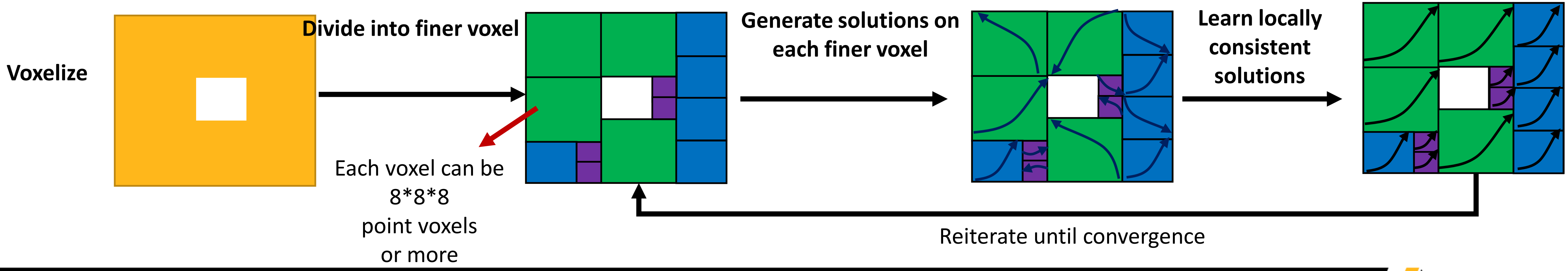




A new ML Solver in latent space

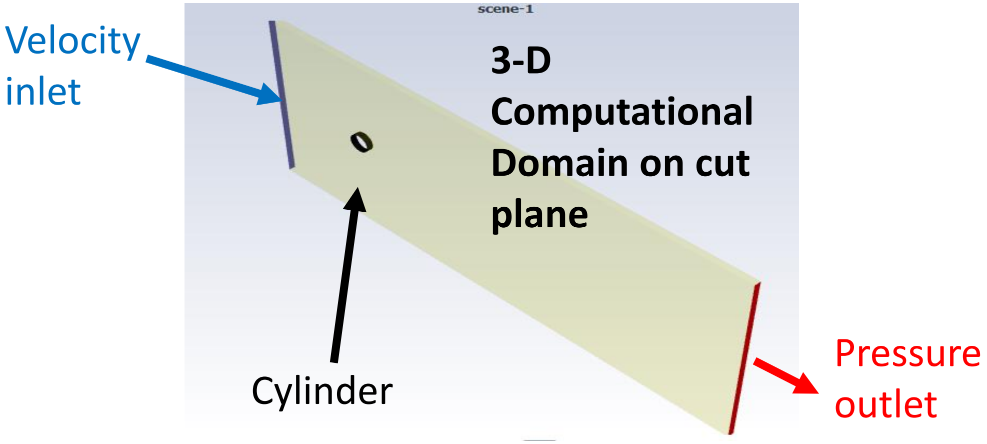
- Motivation:
  - Geometries and physics have lots of patterns!
  - Do we need to solve from scratch?
- Key insights:
  - Generate solutions, *a priori,* at a coarser resolution
  - Breaks and learns solution on a finer resolution
  - Assembles the finer solutions
  - Scalable: geometries or complex physics
  - 50-100x speedup

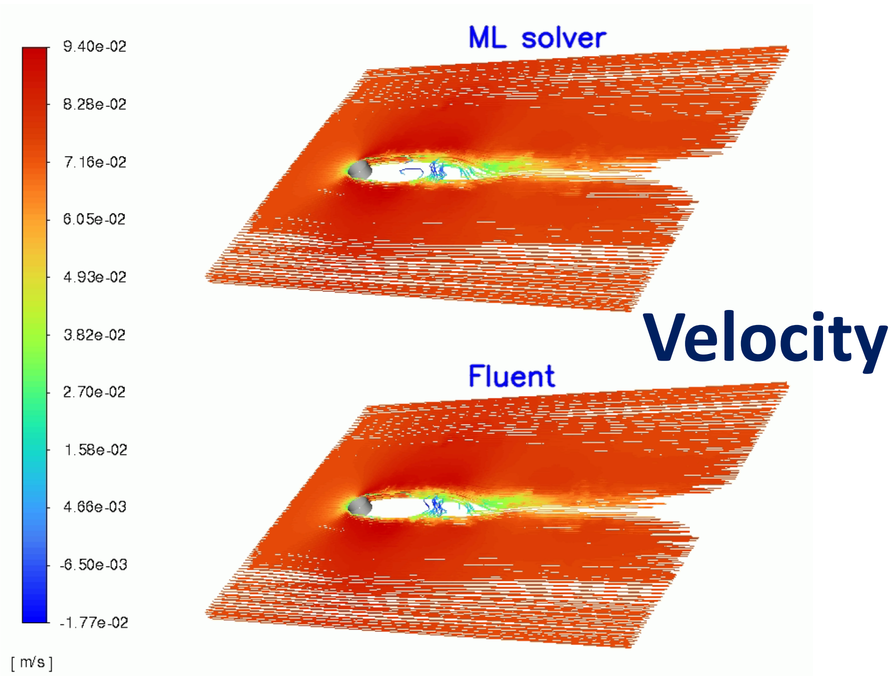
- Publications:
  - CMAME journal: DiscretizationNet: https://arxiv.org/abs/2005.08357
  - ICLR workshop on numerical methods: https://simdl.github.io/files/22.pdf
- Use-cases:
  - Static & Transient PDEs
  - Lid driven cavity and external flows over objects of different shape
  - Industrial Usecase: System level chip thermal simulation tightly coupled CHT with natural convection

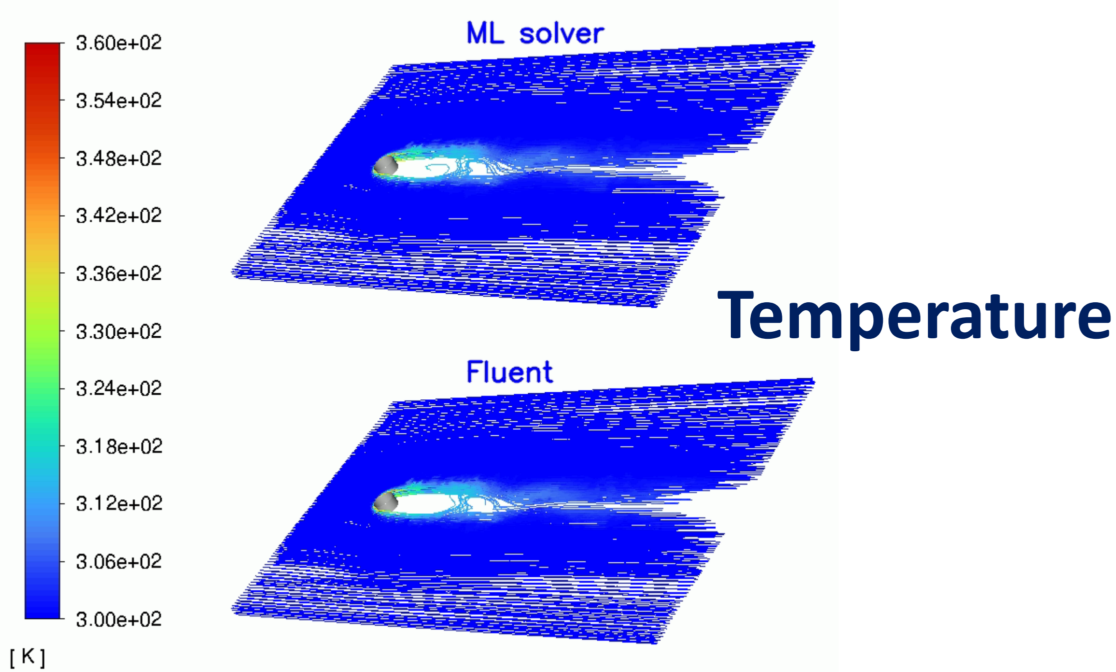


ML Solver: Transient PDEs

- Flow over a cylinder, unsteady RANS
  - Flow Reynolds number = 200
  - Heated cylinder
  - ML-Solver runs 40x-100x faster



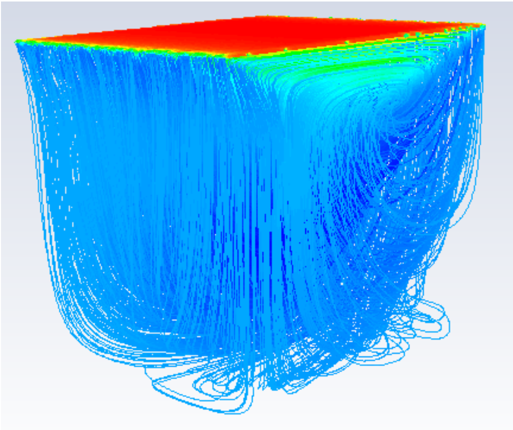


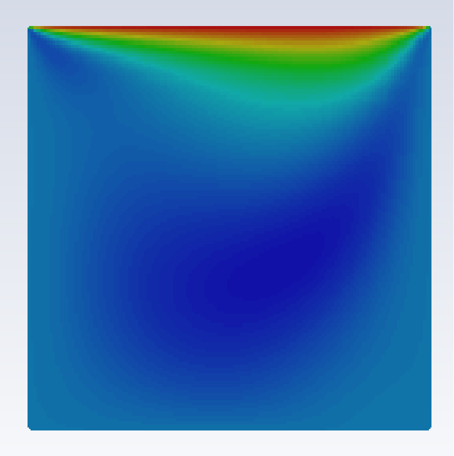


Ansys

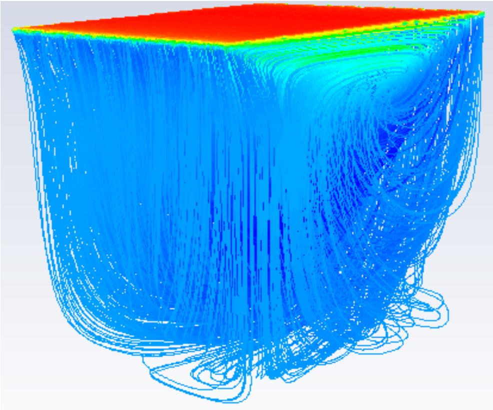
ML Solver: Lid driven cavity results in Fluent

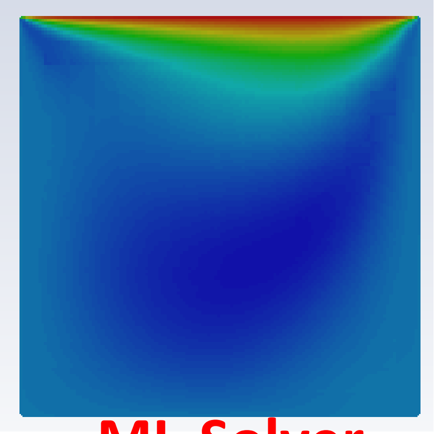
Velocity Streamlines | **Center-plane velocity contour**





Ansys Fluent | **Ansys Fluent**





ML-Solver | **ML-Solver**

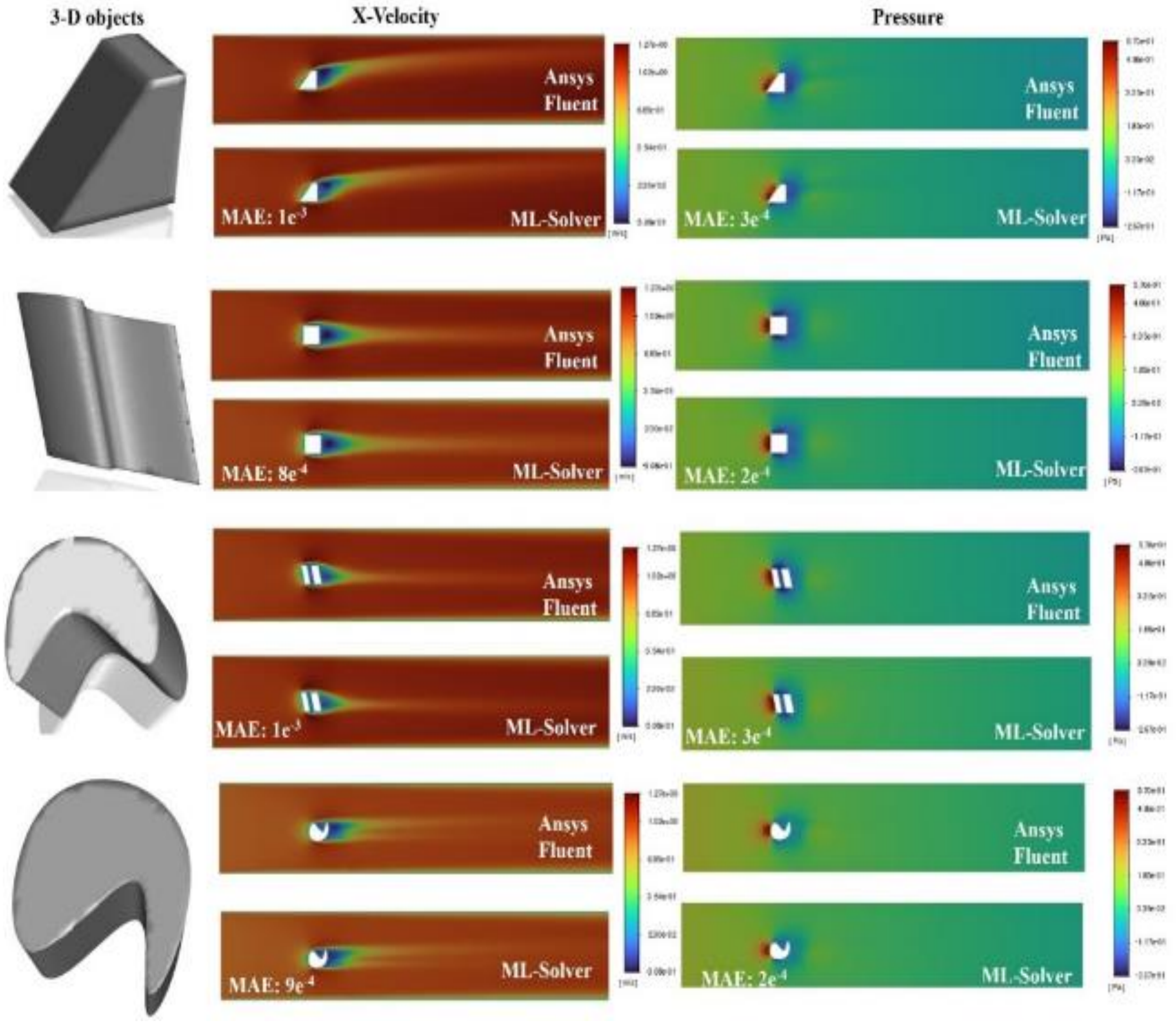
- Results for lid driven cavity with top wall moving at constant velocity
- Verification for unseen cases of Re = 200
- **Speed comparison below for 2.2 million cells using 1 CPU**
- **Time per iteration**
  - ML-Solver: 1.8 seconds
  - Ansys Fluent: 40 seconds
- **Overall convergence**
  - ML-Solver: 36s
  - Ansys Fluent: 5000s
- **Acceleration:**
  - x139

ML Solver: External flow over objects

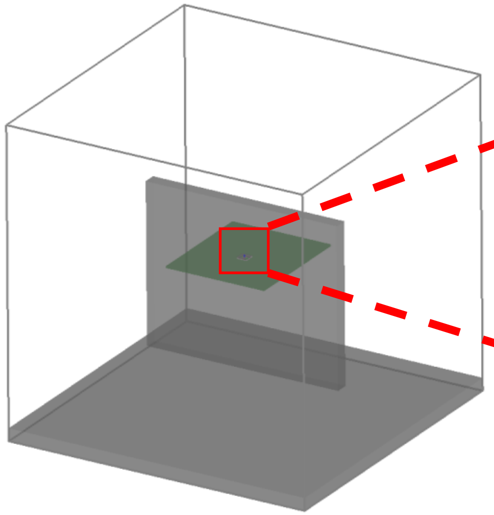
- **Goal:** Develop ML-Solver in the reverse mode (adjoint solver) to optimize engineering designs

- Forward solve using ML-Solver works well
  - Flow over arbitrary objects

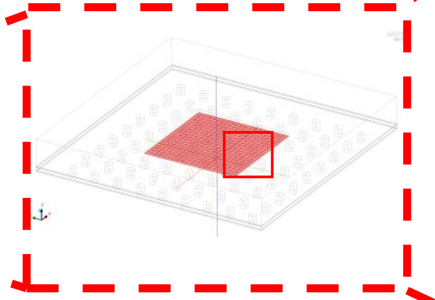
- Adjoint mode needs more work!



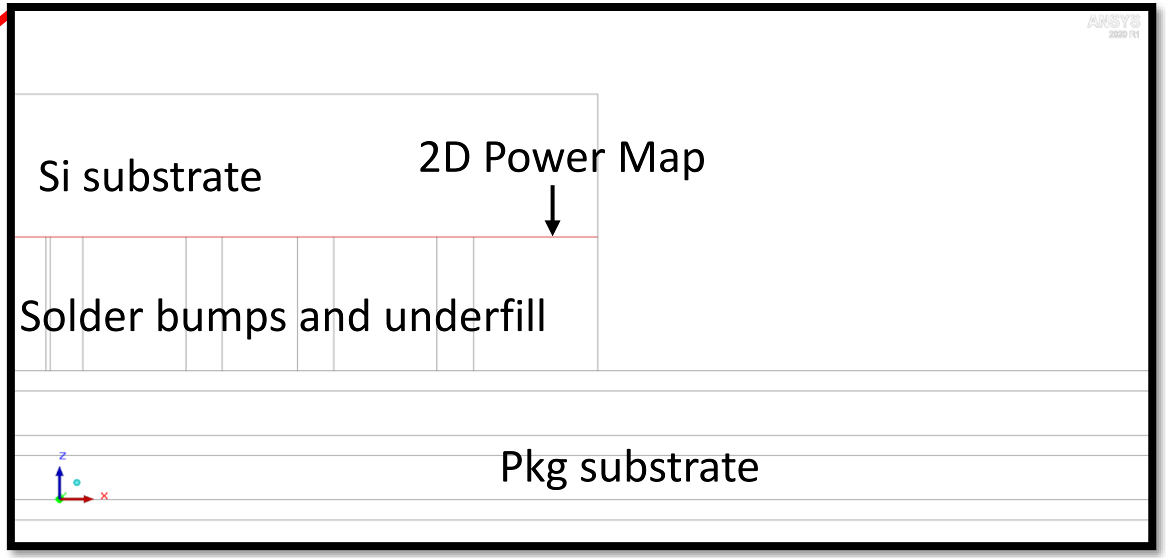
ML Solver: Electronics chip packaging simulation



System level model (~0.16^3 m^3)

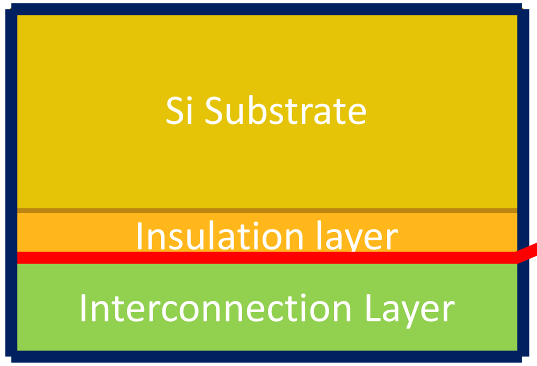


Chip package (~0.032^3 m^3 with heat sink)

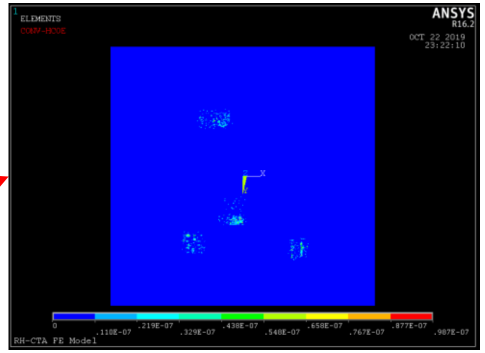


Package cross section

- Tightly coupled temperature and velocity coupled CHT with natural convection
- One way coupling, forced convection
  - Large eddy simulation to capture turbulence



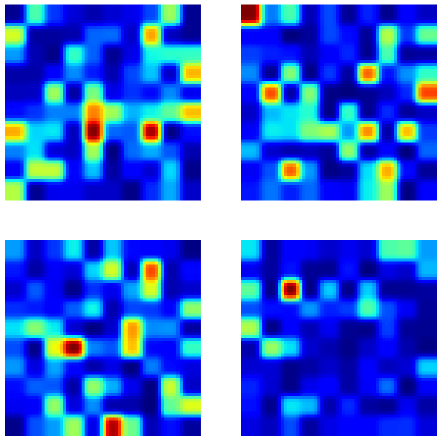
Simplified 3-layer model



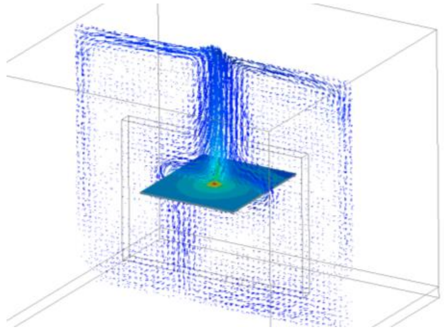
Power map example

ML Solver: Electronics chip packaging simulation, Problem setup

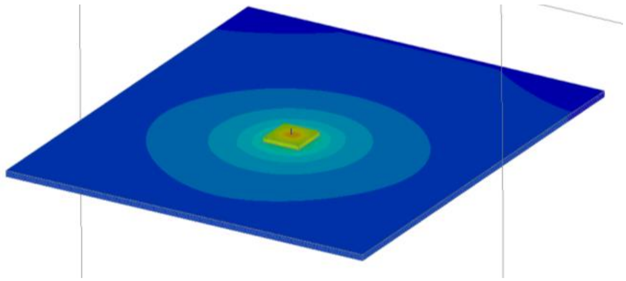
- A small subset of system variables: **Power map**, ambient temperature, material properties(K, Cp, rho) of Die, package, PCB, heat sink geometry(base and fin thickness, # of fins), velocity etc.
- Power map: 4 million sources (10um x 10um) for a 2cm x 2cm chip, each source can take continuous values in given range (e.g., 0 to 10mWatt)
- Physics involved: Heat transfer in solid and Fluid, Navier–Stokes's equations …
- Goal: Given any power map applied on a chip, predict the solutions for *T, u, v, w, p*
- Expected ground truth solutions (*T*)



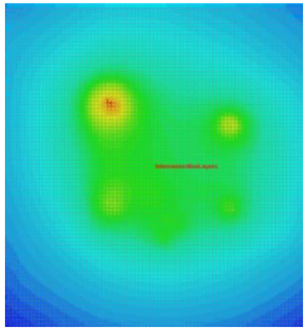
Sample power maps (representatives of real applications)



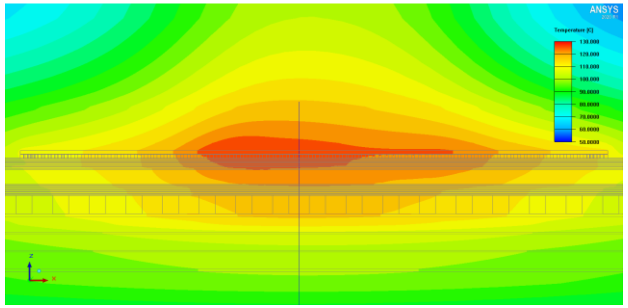
System



PCB



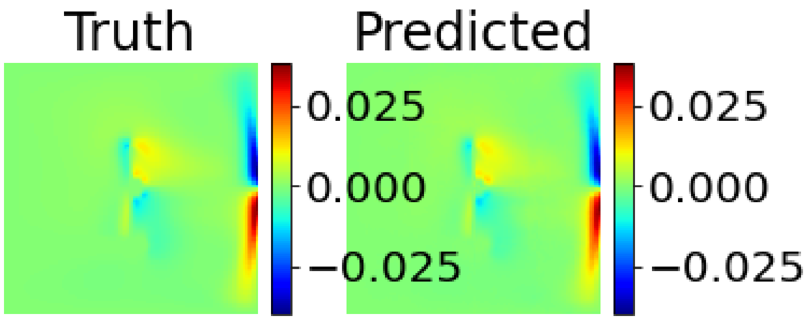
Die top



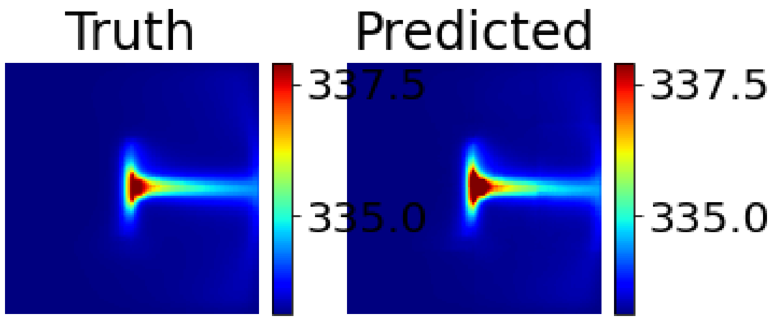
Die cut plane

ML Solver: Electronics chip packaging simulation, preliminary results

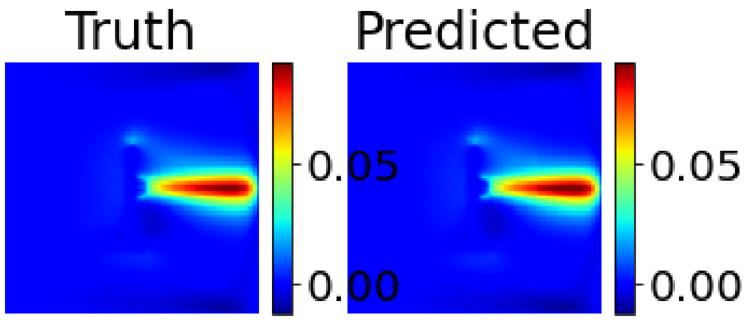
- ML solver prediction for one test case



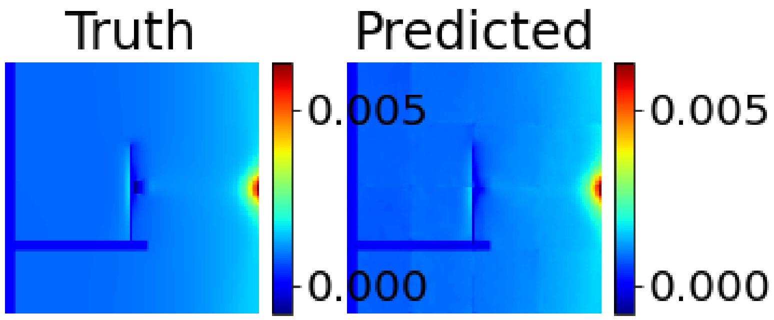
X-velocity



Temperature

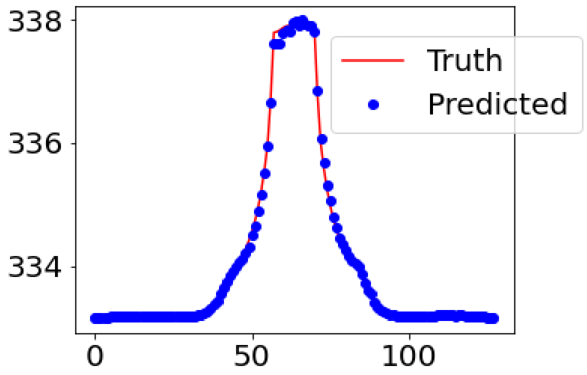


Z-velocity

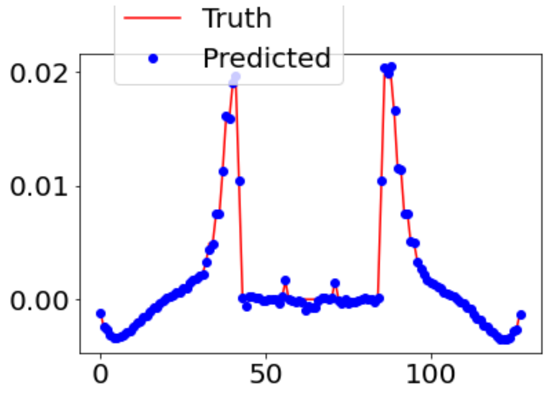


Pressure

- Line plot comparison for the same case



Temperature at center plane



Velocity at center plane

Thermal Artist, Fast architecture level what if thermal analysis

Fast Transient Thermal Solving for Architecture Level What-if Analysis

Physical data (Chip + block bbox)

Transient Power Scenario *(from Emulator/PA or w/ block functional Modes)*

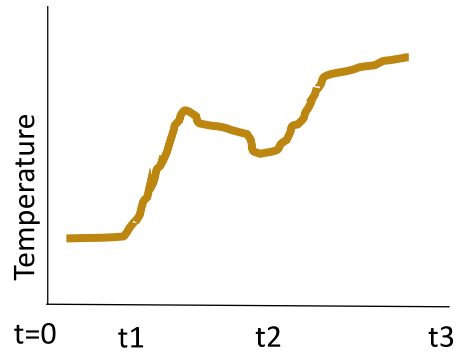
Selected BC from Estimated Pkg/Brd

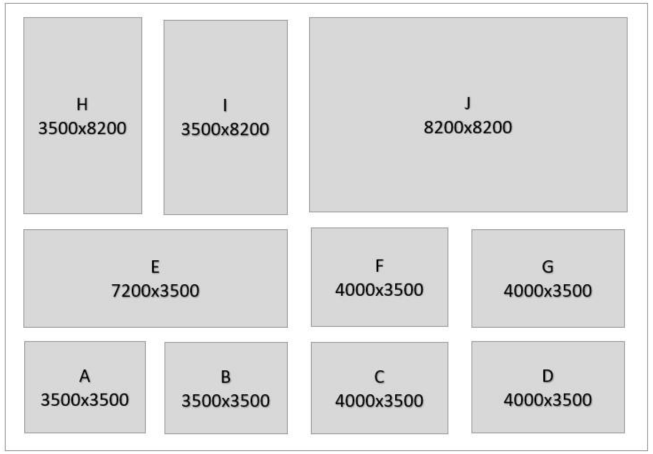
Block Power Model

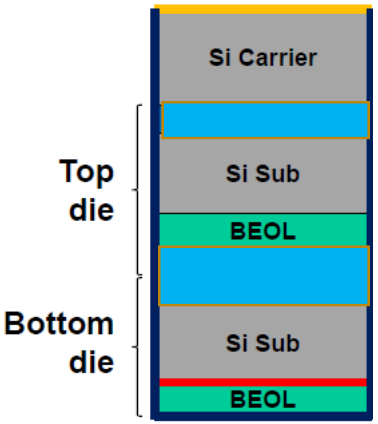
Architecture What-if

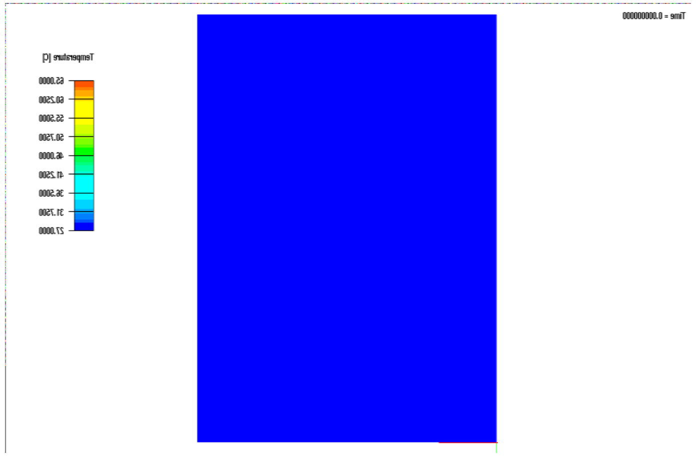
Steady State/ Transient Thermal profiles

Reporting : Top Ranked Thermal Hotspot in Time





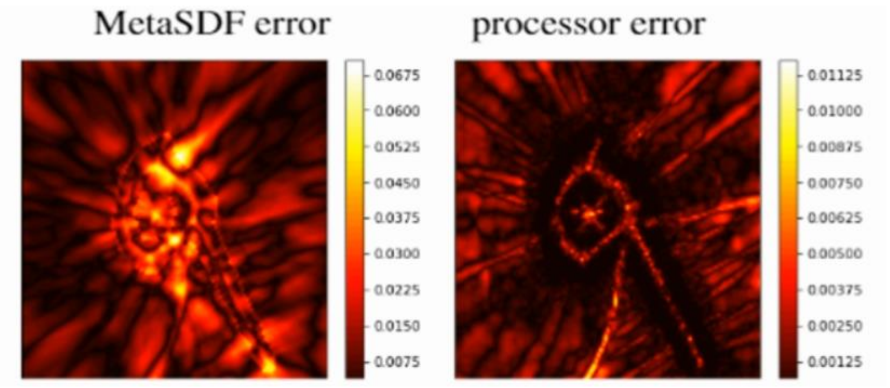




"Model-based Digital Twin for Anomaly Detection of On-chip Transient Thermal Response", A. Kumar, N. Chang, et al., DesignCon 2021

Encoding geometry

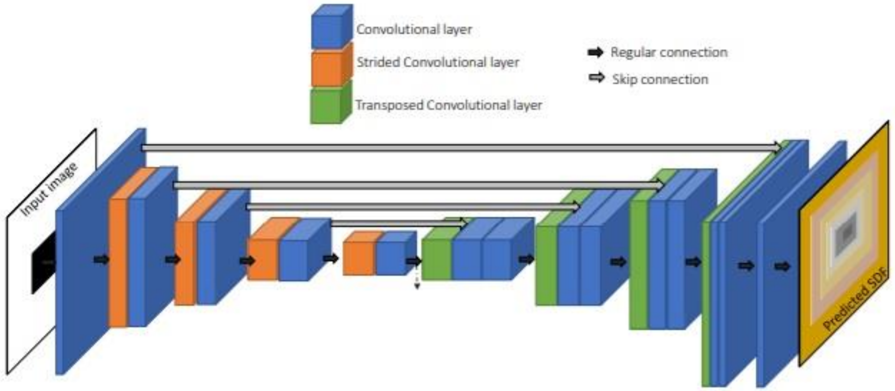
- Identified constraints of encoding for numerical simulations
- Improved the state of the art from research community
- Generalization: Trained to SpaceClaim simple parts scaled to unseen 2D/3D geometries

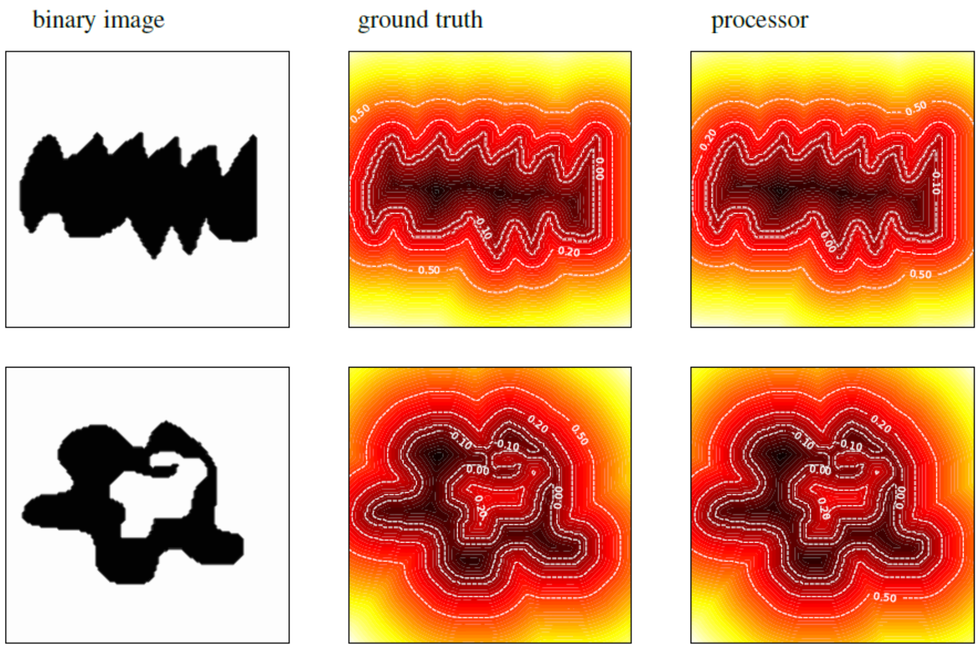


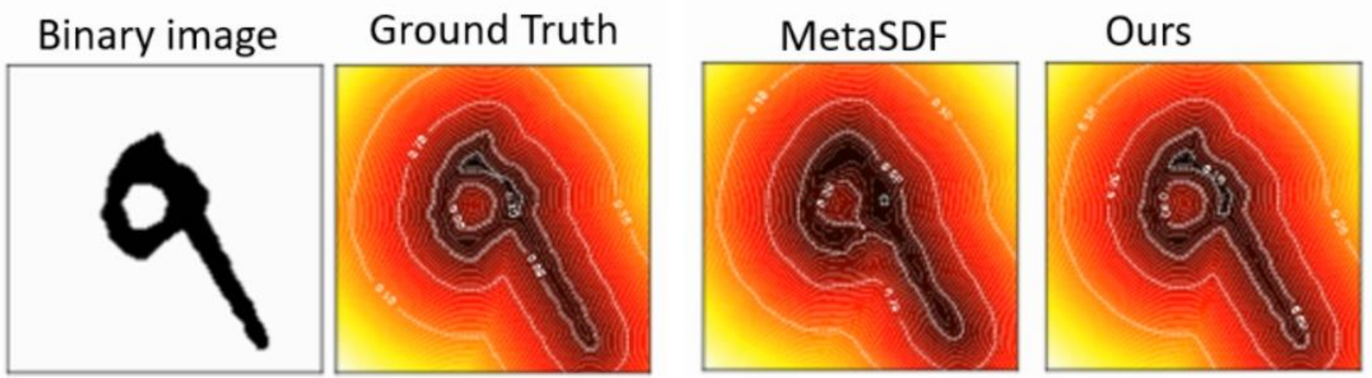
Accepted at the ICLR 2021 Workshop on Geometrical and Topological Representation Learning

GEOMETRY ENCODING
FOR NUMERICAL SIMULATION

Amir Maleki[1], Jan Heyse[2], Rishikesh Ranade[1], Ha
[1]Ansys Inc, [2]Stanford University
{amir.maleki, rishikesh.ranade, haiy
heyse@stanford.edu

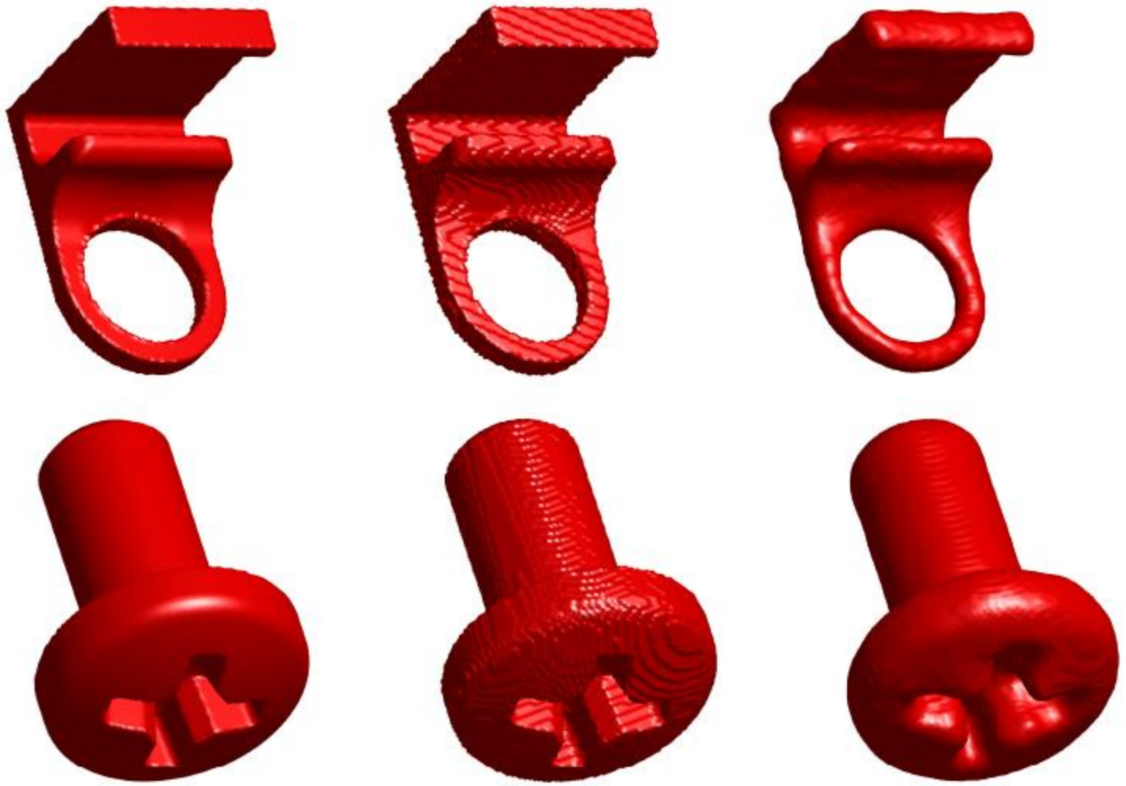


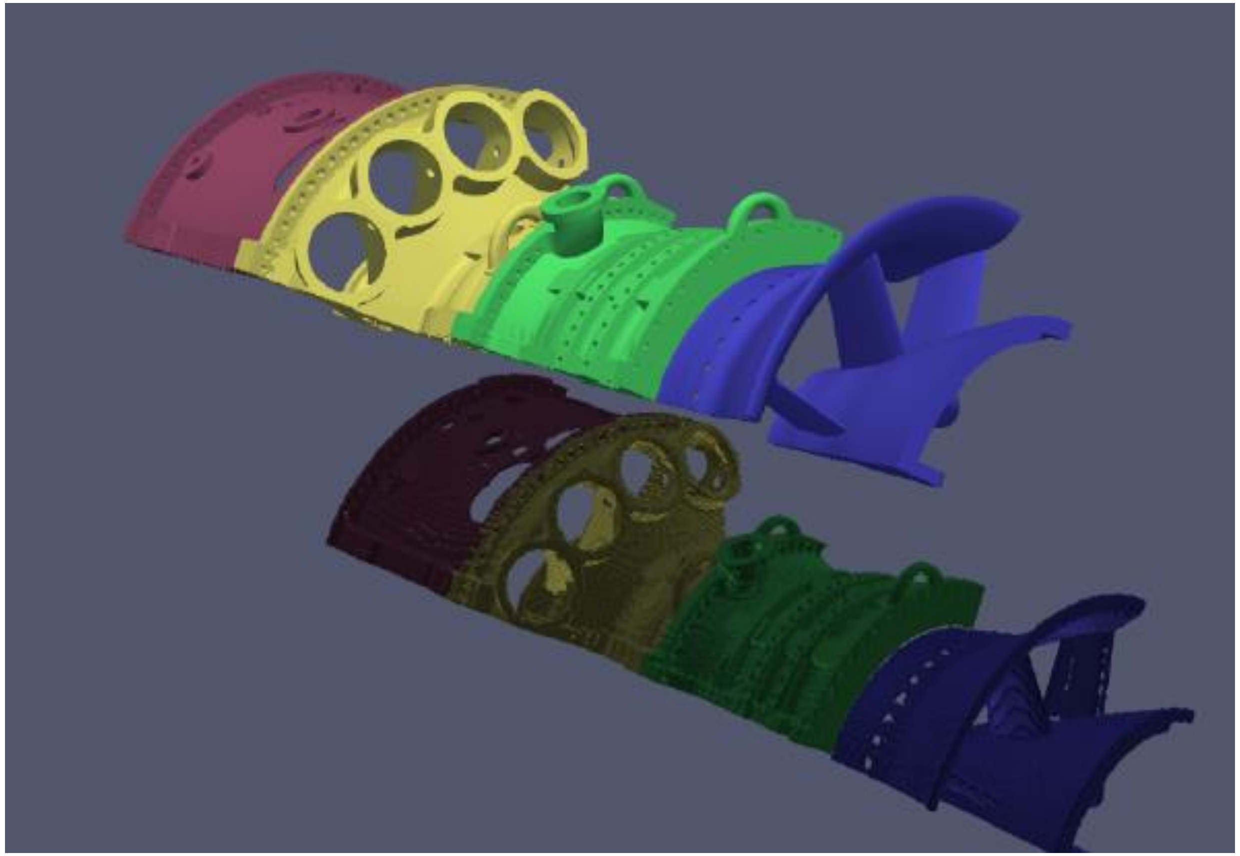




Encoding geometry: Latest 3D Results

Ground truth | Reconstructed surface | Reconstructed surface with compression

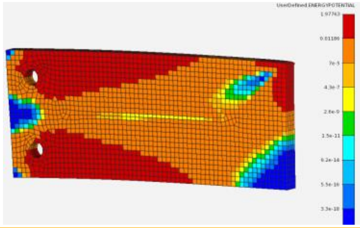




Ground truth (top) and reconstruction (bottom) of a GE turbine engine

Learning topology optimization

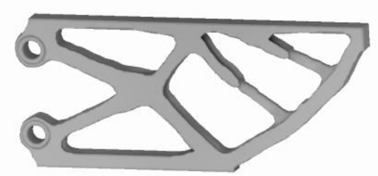
- Motivation
  - Topology optimization
    - Numerical technique to distribute the material inside a design region to have optimal properties and satisfies the prescribed loads, design and manufacturing constraints
    - Traditionally, solvers developed since 1989 [1] use minimization of elastic strain(called SIMP) energy of a part for a given total weight and boundary conditions specified.
  - The technique is general purpose and scales to other physics but extremely time consuming for full fidelity.
- Learning the relationship between the strain energy distribution and evolving shapes
  - 3d dataset available with evolving densities and final density for various configuration of loading & geometries
  - Challenges: Generalization to new geometries outside of distribution



ElemDensity = f(ElemStrainEnergy)

Learn f by Deep Learning

s.t: some protected locations




Engineering Applications of Artificial Intelligence
Volume 106, November 2021, 104483

Algorithmically-consistent deep learning frameworks for structural topology optimization

Jaydeep Rade, Aditya Balu, Ethan Herron, Jay Pathak, Rishikesh Ranade, Soumik Sarkar, Adarsh Krishnamurthy

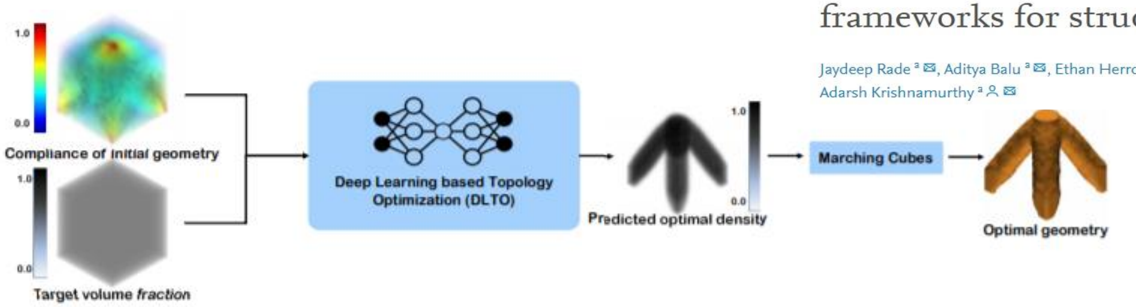


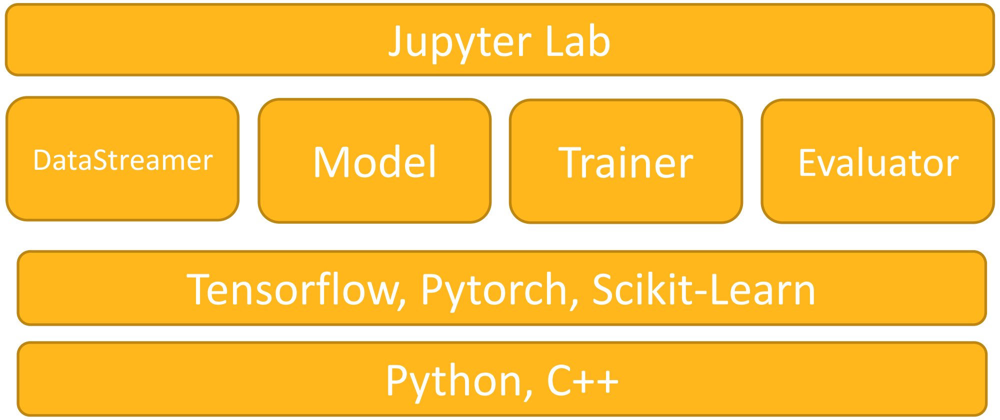
Figure 1: ***Overview:*** *We propose a deep learning based topology optimization framework. The input to this framework is the compliance of the initial geometry along with the target volume fraction. Using the DLTO framework, we predict the optimal density of the geometry without any requirement of iterative finite element evaluations. We then convert the predicted optimal density of the geometry and convert it into triangular surface mesh representation using the marching cubes algorithm to give the final optimal design geometry.*

https://arxiv.org/pdf/2012.05359.pdf

New ML Framework

Accelerate ML dev productivity

- High level API for managing training, ML models, debugging, visuals, and evaluation
- Thin web-client, Jupyter Labs style
- Use from a library of templates
- Create images based on a base docker image
- Tensorflow/PyTorch backends
- C++ and Python
- Auto-scale training with GPUs in DGX cluster
- Embed ML Dlls into product or use ML as web service



Imports Components

```
import MLFramework.MLFramework.components.trainer.component.Trainer as Trainer
import MLFramework.MLFramework.components.data_streamer.component.DataStreamer a
import MLFramework.visualize.vis_tools.plot3D as plt
```

Interactive 3D Data Visualization Tools

```
fig = plt.make_3d_figure_from_data(obj_data2)
fig.show()
```

DataStreamer API

```
datastreamer = DataStreamer(file_path, inputs=input_cols, outputs=output_cols)
train_loader, val_loader = datastreamer.train_data, datastreamer.val_data
```

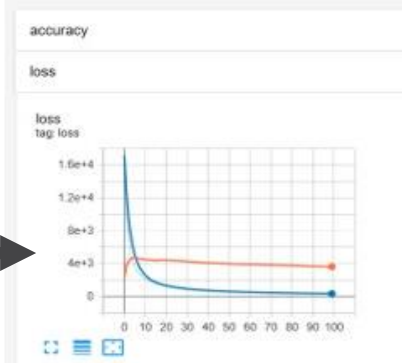
Model API

```
model = utils.get_model(input_dim=11, output_dim=4, model_name='nn1')
```

Trainer API

```
my_trainer = Trainer(model=model, run=1, data_loader=train_loader,
                     validation_loader=val_loader, output_dir="output")
my_trainer.train()
```

93% 2134/2300 [05:49<00:19, 8.64it/s, loss=3.66e+01]



Hybrid digital twin

Motivation:

- Parameter calibration for equipment health monitoring
- Difficult to model system degradation using only physics alone
- Can we use the real-time data collected to reduce the residue in physics-based modelling?

Methodology:

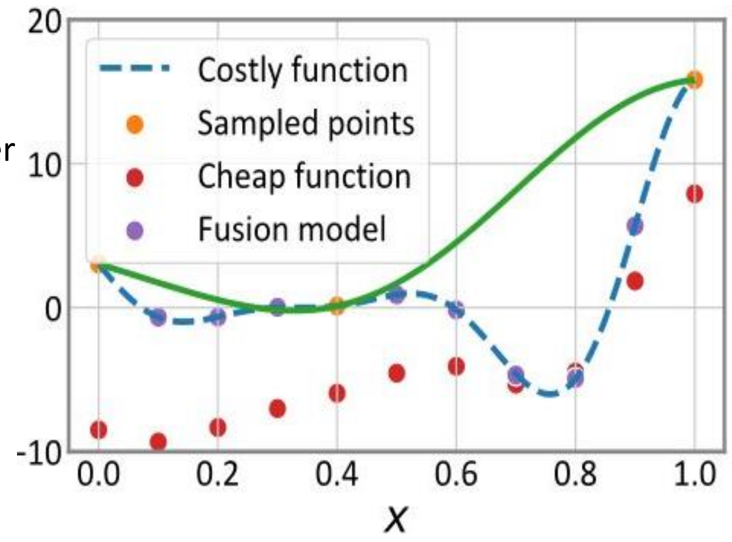
- Use system characteristics to create a physics model
- Estimate the parameters using physics model
- Integrate the estimation with data-based model to get more accurate parameter calibration

Current projects and contribution:

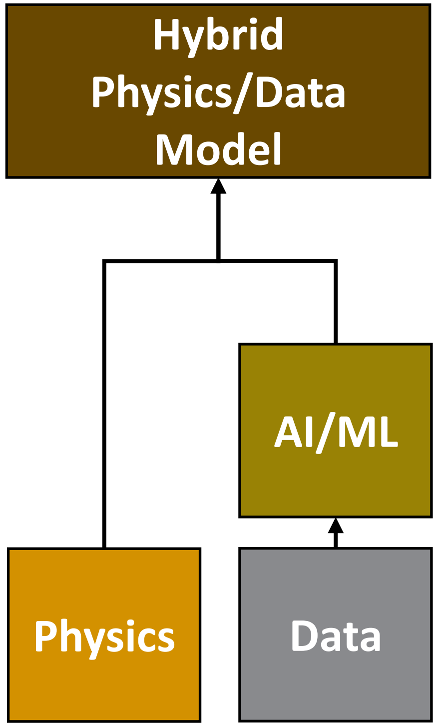
- Flowserve: Delivered noise fitting module. Working on Fusion model algorithm (More accurate functionality based on Machine Learning algorithm)
- Boeing: Preprocessing the raw data. Establishing baseline using Deep Learning based Neural Net models

Future directions:

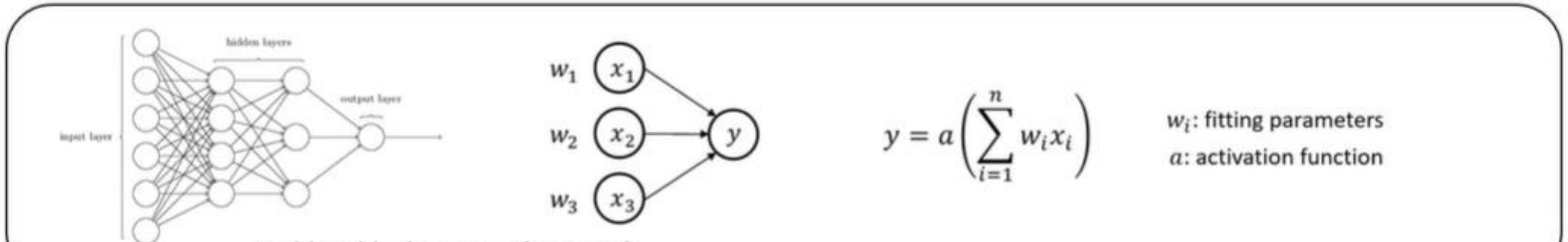
- Software integration for internal release
- Deploy these models as cloud-based microservice to the customers
- Add more functionality and improved models in subsequent updates



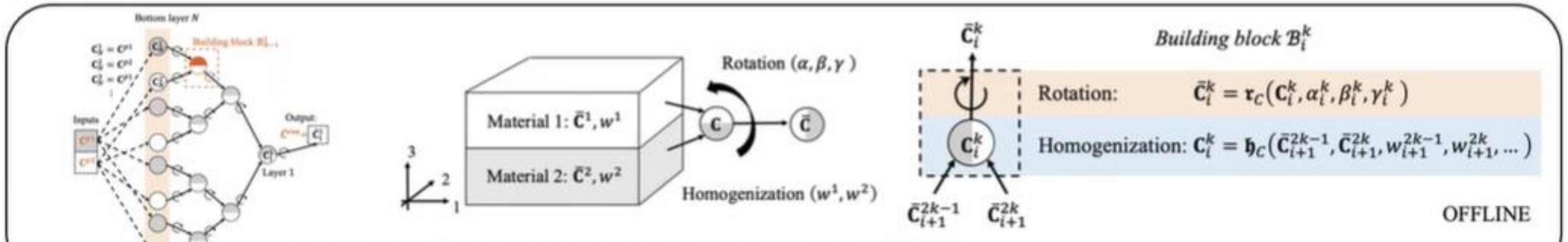
Full Integration



Deep Material Network



Building block in **neural network**

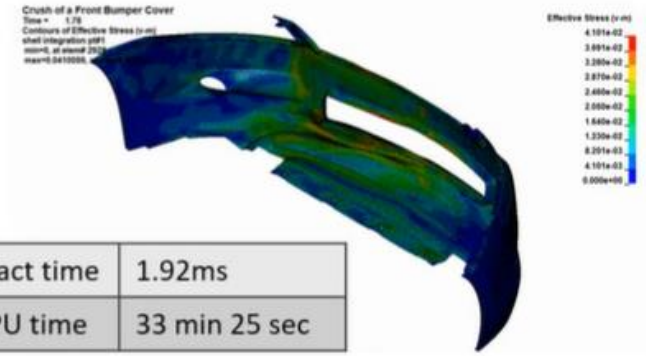


Mechanistic building block in **deep material network (DMN)**

- New ML based network to handle multi-scale composite materials
- Fixes material history dependencies without loss of physics
- Several orders(1000x) faster than current model based multi-scale method

Multiscale Material Modeling using

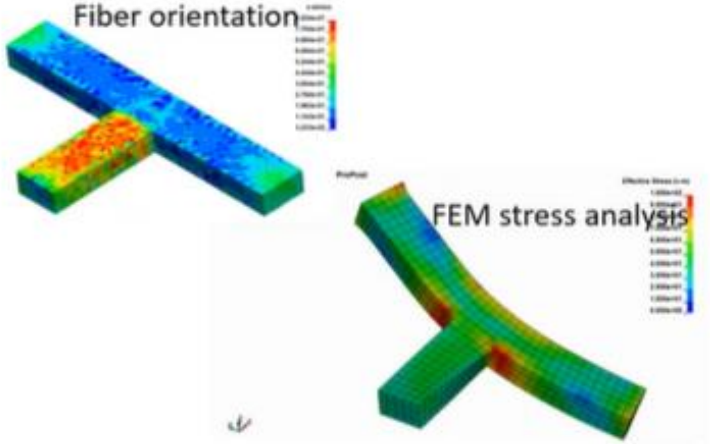
DMN in FEM Structure Analysis



Impact time	1.92ms
8 CPU time	33 min 25 sec

25256 shell elements, 25600 nodes

LS-DYNA DMN Package



Ansys

Additive Manufacturing, ML accelerated fast thermal strain

Thermal Strain can take days or weeks to simulate a realistic part—which is not feasible for most customers. A machine learning emulator proof-of-concept has been developed that:

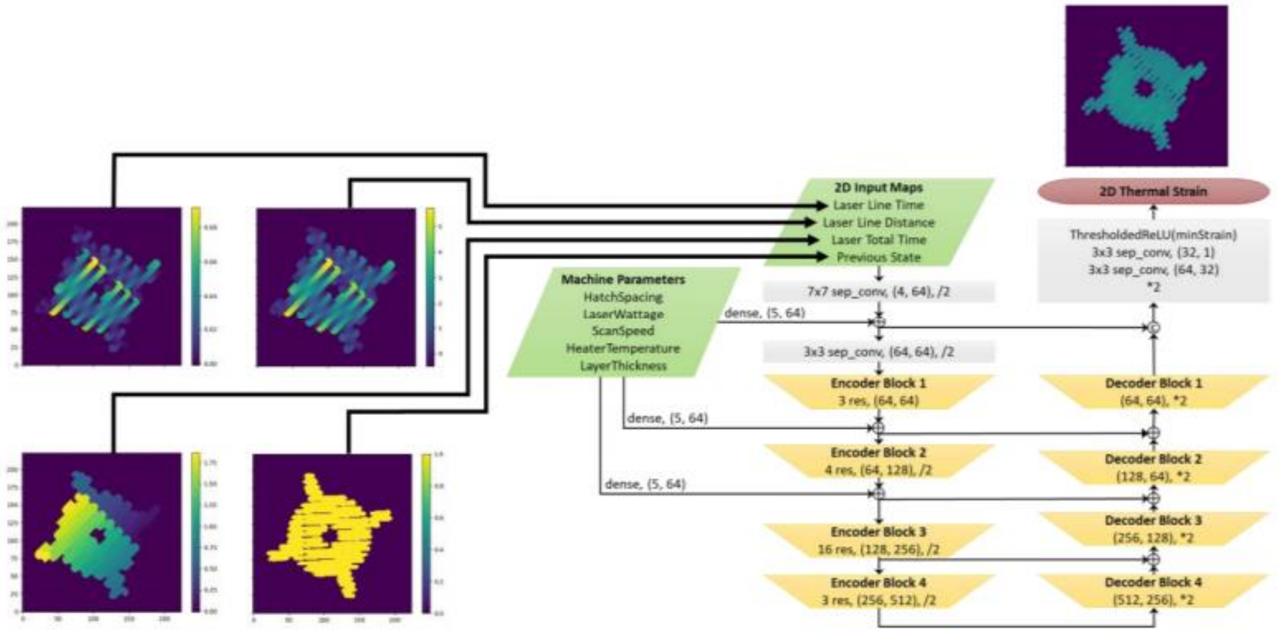
Generalizes on machine parameters	Generalizes on geometry	Uses a deep convolutional neural network to achieve 1 to 2 orders of magnitude speedup

Single Bead Bounds

Parameter	Range	Units
Laser Wattage	50 – 700	W
Scan Speed	350 – 2500	mm/s
Layer Thickness	10 – 100	µm
Beam Diameter	20 - 140	µm
Heater Temperature	20 – 500	C

Slicer Bounds

Parameter	Range	Units
Hatch Spacing	60 – 1000	µm
Starting Angle	0 – 180	deg
Rotation Angle	0 – 180	deg
Stripe Width	1 – 100	mm



Ansys