

TSINGHUA-PRINCETON-COMBUSTION INSTITUTE

2026 SUMMER SCHOOL ON COMBUSTION

AI for Combustion

Matthias Ihme

Stanford University, USA

July 08-10, 2026



TSINGHUA-PRINCETON-COMBUSTION INSTITUTE

2026 SUMMER SCHOOL ON COMBUSTION

Key Activities / 重要活动			
July 5 (Sunday) /7 月 5 日 (周日)	10:00-17:30	Registration 注册	Northeast Gate, Lee Shau Kee Sci. and Tech. Building 李兆基科技大楼东北门
	18:30	Welcome Reception 开班仪式	A-278, Multifunction Room, Lee Shau Kee Sci. and Tech. Building 李兆基科技大楼多功能厅
Class Schedule / 课程安排			
Schedule 时间	Location 地点	Morning 上午 (9:00-9:50/10:00-10:50/11:00-11:50)	Afternoon 下午 (14:00-14:50/15:00-15:50/16:00-16:50)
July 6-10 (Monday-Friday) /7 月 6-10 日 (周一至周五)	Zone A, 6 th Teaching Building 第六教学楼 A 区	Theoretical and Numerical Combustion classroom: 6A018 (0th floor) Thierry Poinsot	Combustion Chemistry: From Fundamentals to Kinetic Modelling for Low-Carbon Technologies classroom: 6A018 (0th floor) Alison Tomlin
July 6-7 (Monday-Tuesday) /7 月 6-7 日 (周一至周二)	Zone A, 6 th Teaching Building 第六教学楼 A 区	Spectroscopic Diagnostics for Combustion Chemistry classroom: 6A203 (2nd floor) Pascale Desgroux	Quantum Mechanics, Statistical Mechanics, and Machine Learning for Molecular Simulations classroom: 6A203 (2nd floor) Alexandre Tkatchenko
July 8-10 (Wednesday-Friday) /7 月 8-10 日 (周三至周五)	Zone A, 6 th Teaching Building 第六教学楼 A 区	Introduction to Plasma-assisted Combustion classroom: 6A203 (2nd floor) Deanna Lacoste	AI for Combustion classroom: 6A016 (0th floor) Matthias Ihme
Special Activities / 特殊活动			
July 5 (Sunday) /7 月 5 日 (周日)	13:30-17:30	Art Museum Visit / 艺术博物馆参观	Tsinghua University Art Museum 清华大学艺术博物馆
July 7 (Tuesday) /7 月 7 日 (周二)	17:00-18:00	Campus Tour / 校园游览	Tsinghua University 清华大学

July 8 (Wednesday) /7 月 8 日 (周三)	17:00-17:30	Group Picture Taking / 暑期学校合影	Mong Man Wai Concert Hall 蒙民伟音乐厅
July 8 (Wednesday) /7 月 8 日 (周三)	18:30-19:30 19:30-21:00	Poster Presentation / 海报展示 Career Panel / 职业发展论坛	B-518, Lee Shau Kee Sci. and Tech. Building 李兆基科技大楼 B-518 会议室
July 9 (Thursday) /7 月 9 日 (周四)	18:00	Farewell Reception / 欢送会	Guan Chou Yuan Restaurant 观畴园餐厅
July 10 (Friday) /7 月 10 日 (周五)	8:00-18:00	Program Certificate Distribution / 学习证 书发放	6 th Teaching Building 第六教学楼
July 11 (Saturday) /7 月 11 日 (周六)	9:30-11:30	CCE Laboratory Tour / 燃烧能源中心实验 室参观	Northeast Gate, Lee Shau Kee Sci. and Tech. Building 李兆基科技大楼东北门

Electronic version of all lecture materials are available at the summer school website



AI for Combustion

Matthias Ihme

mihme@stanford.edu

Module 1: Introduction and Prerequisites

Learning goals

- Historical perspective
- ML challenges combustion
- Mathematical foundation and notations

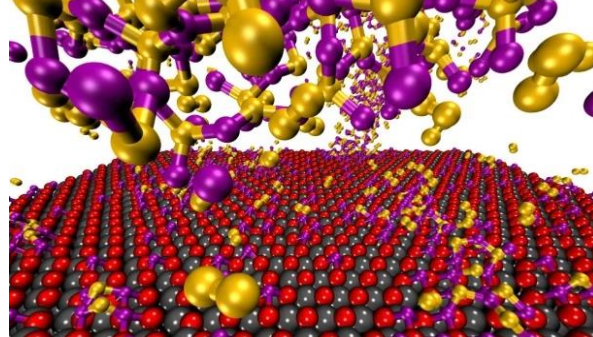
- ML learning problem: Data, Learning task , Loss functions
- Data for combustion science and engineering

Combustion trends

Sustainable Energy Utilization



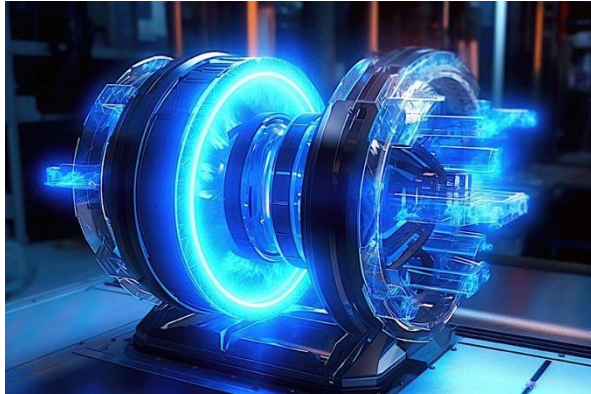
Renewable Fuels



Mitigate Emissions



Advanced propulsion solution



Decarbonize Process Heating

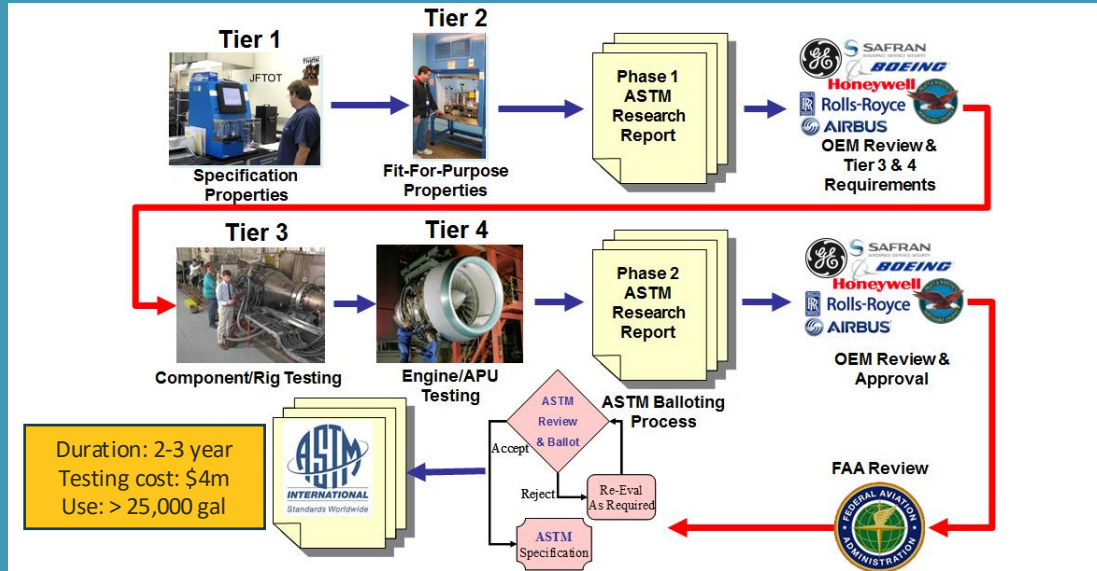


Fire Safety and Hazards



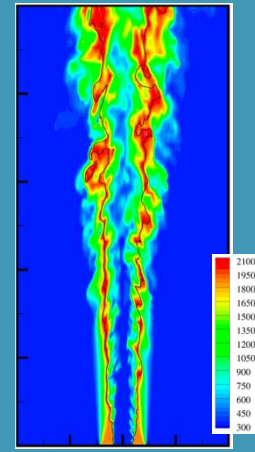
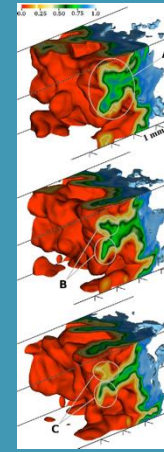
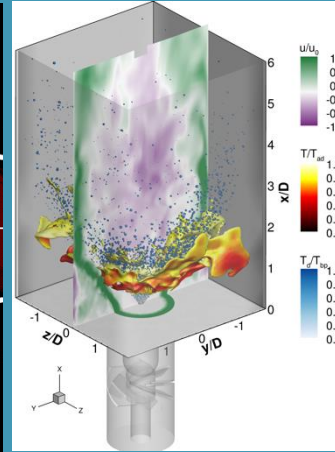
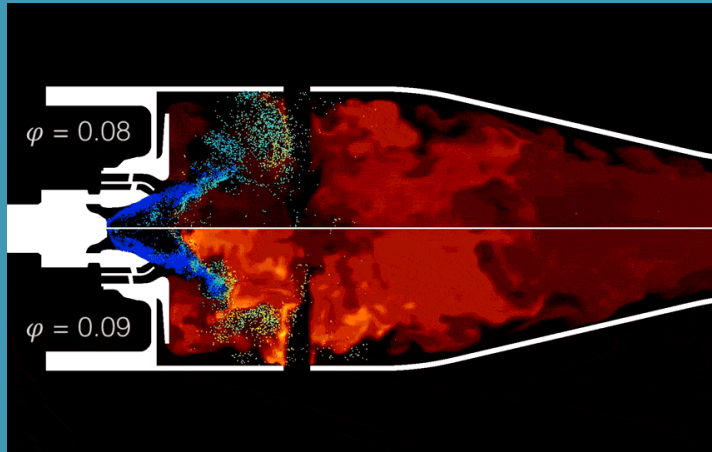
Combustion opportunities

- Accelerate fuel certification and fuel screening



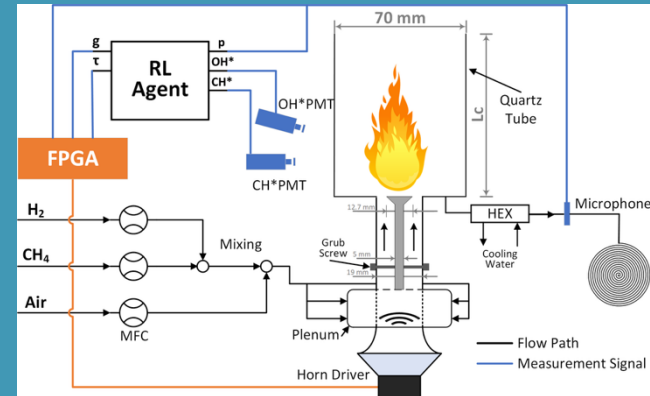
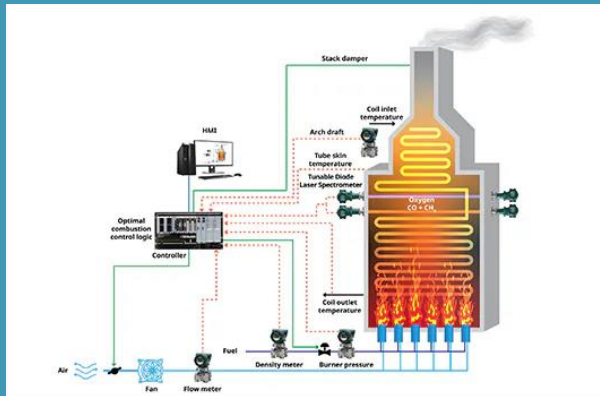
Combustion opportunities

- Accelerate fuel certification and fuel screening
- Predictive modeling and simulation tools



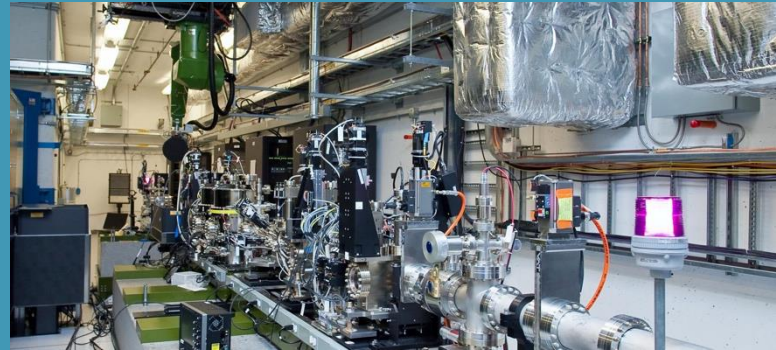
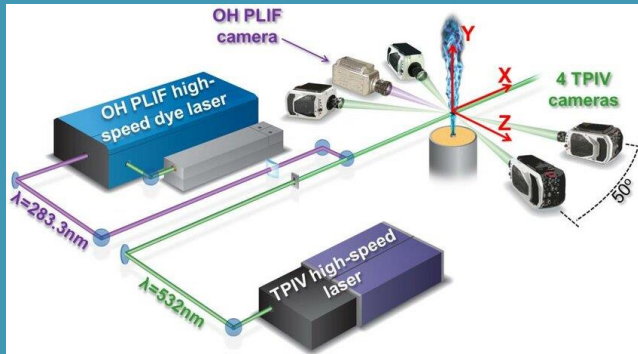
Combustion opportunities

- Accelerate fuel certification and fuel screening
- Predictive modeling and simulation tools
- Novel control and optimization techniques



Combustion opportunities

- Accelerate fuel certification and fuel screening
- Predictive modeling and simulation tools
- Novel control and optimization techniques
- Innovative methods to accelerate data analysis

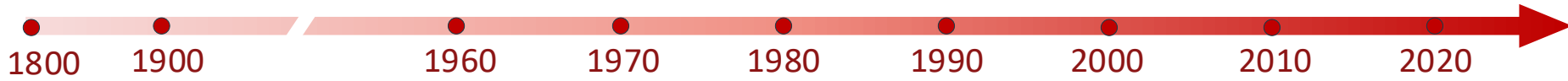


Combustion opportunities

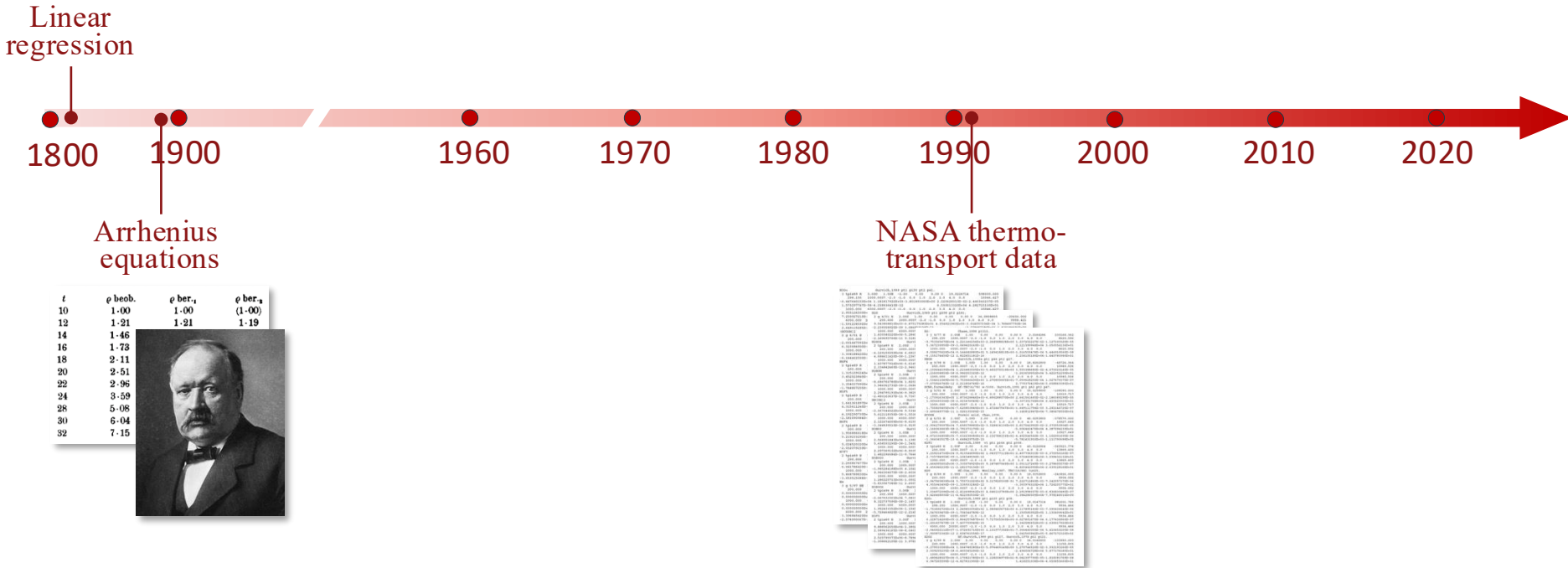
- Accelerate fuel certification and fuel screening
- Predictive modeling and simulation tools
- Novel control and optimization techniques
- Innovative methods to accelerate data analysis

How can AI complement and accelerate combustion science and engineering?

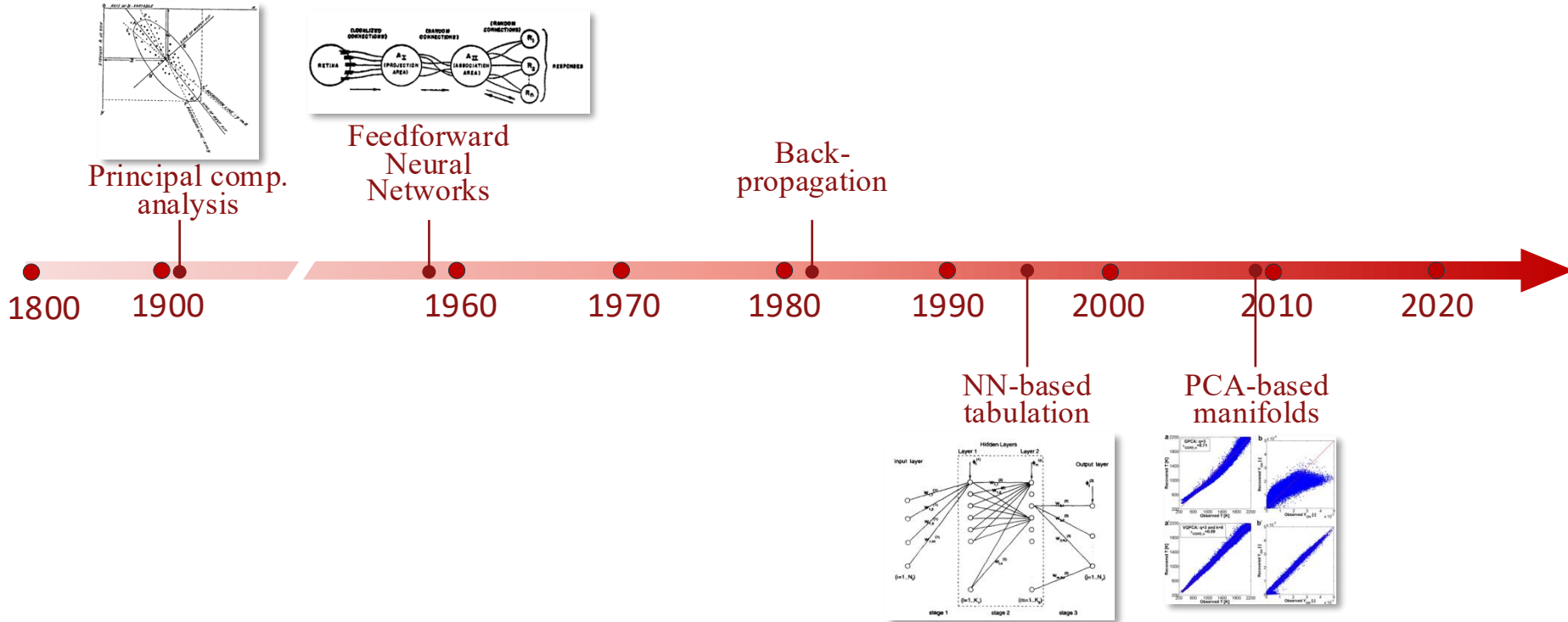
Key milestones in ML and AI



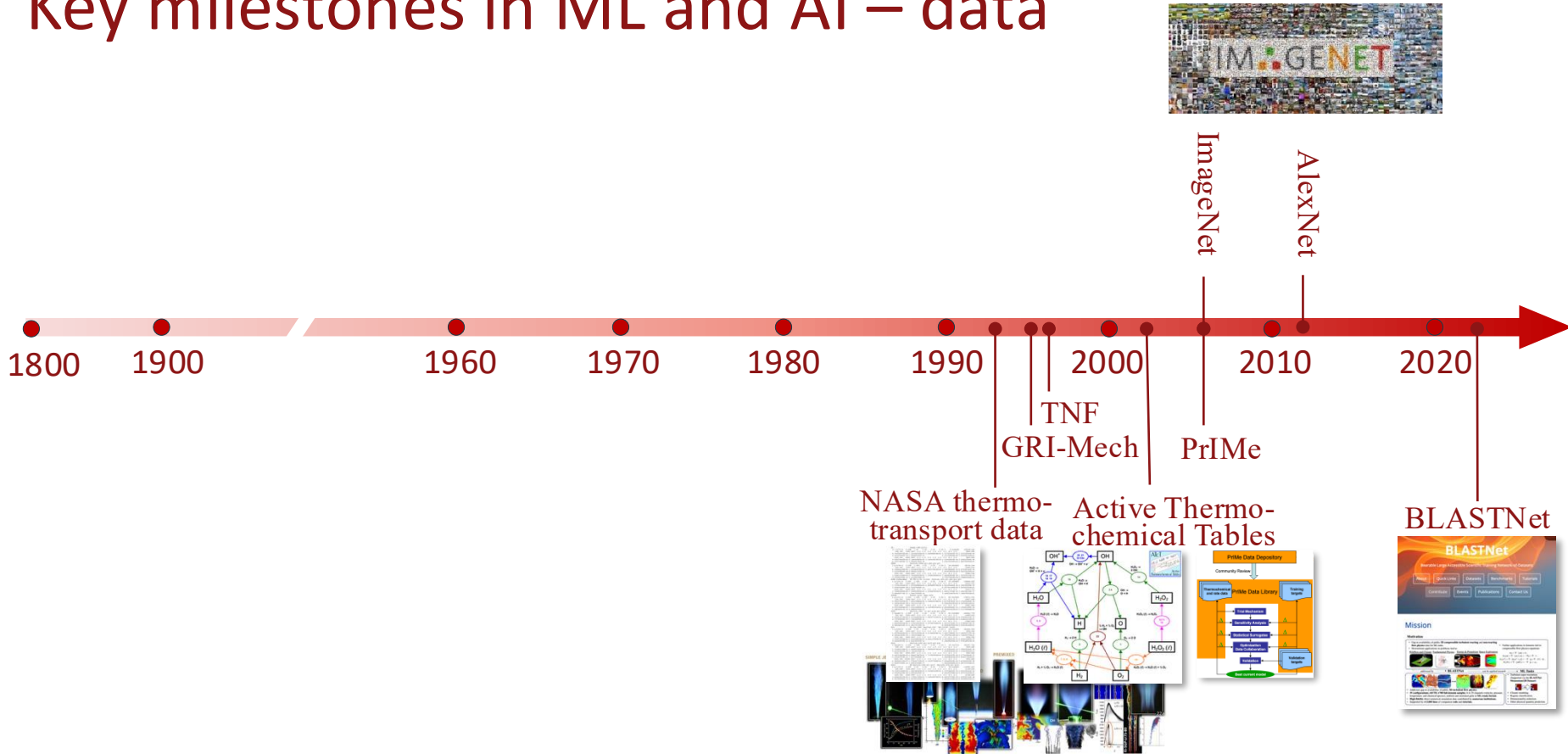
Key milestones in ML and AI – methods



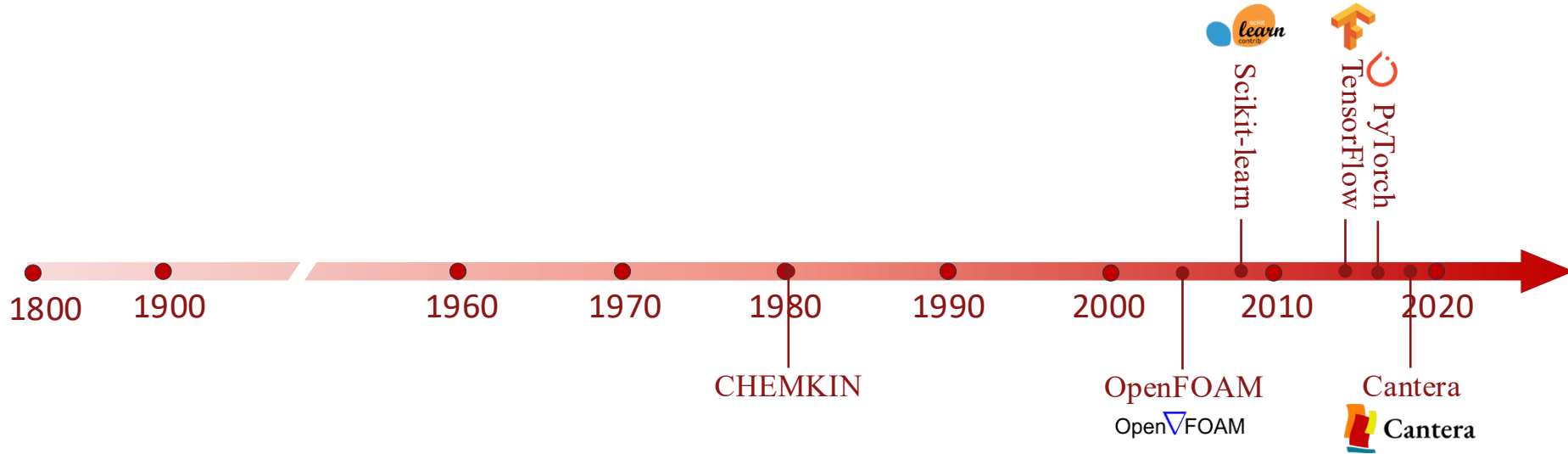
Key milestones in ML and AI – methods



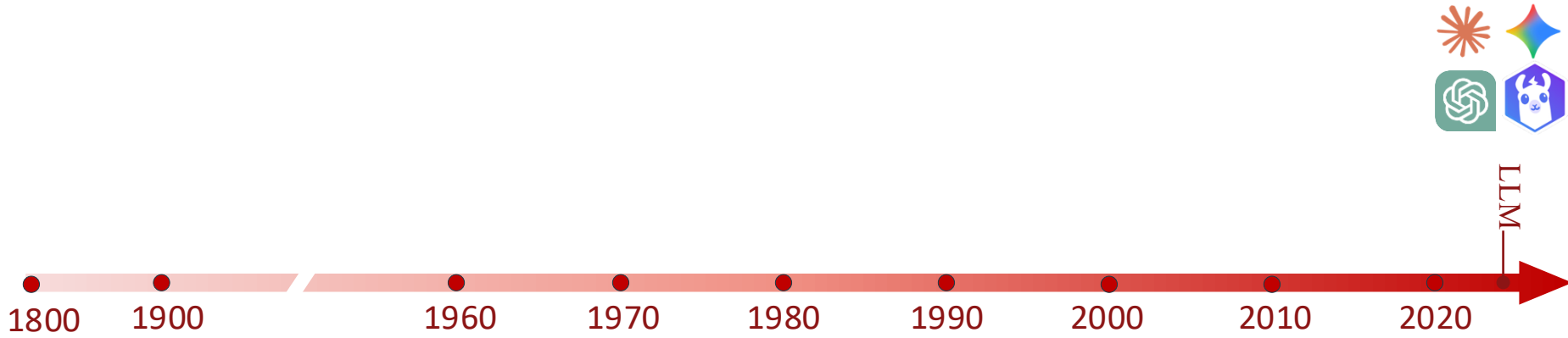
Key milestones in ML and AI – data



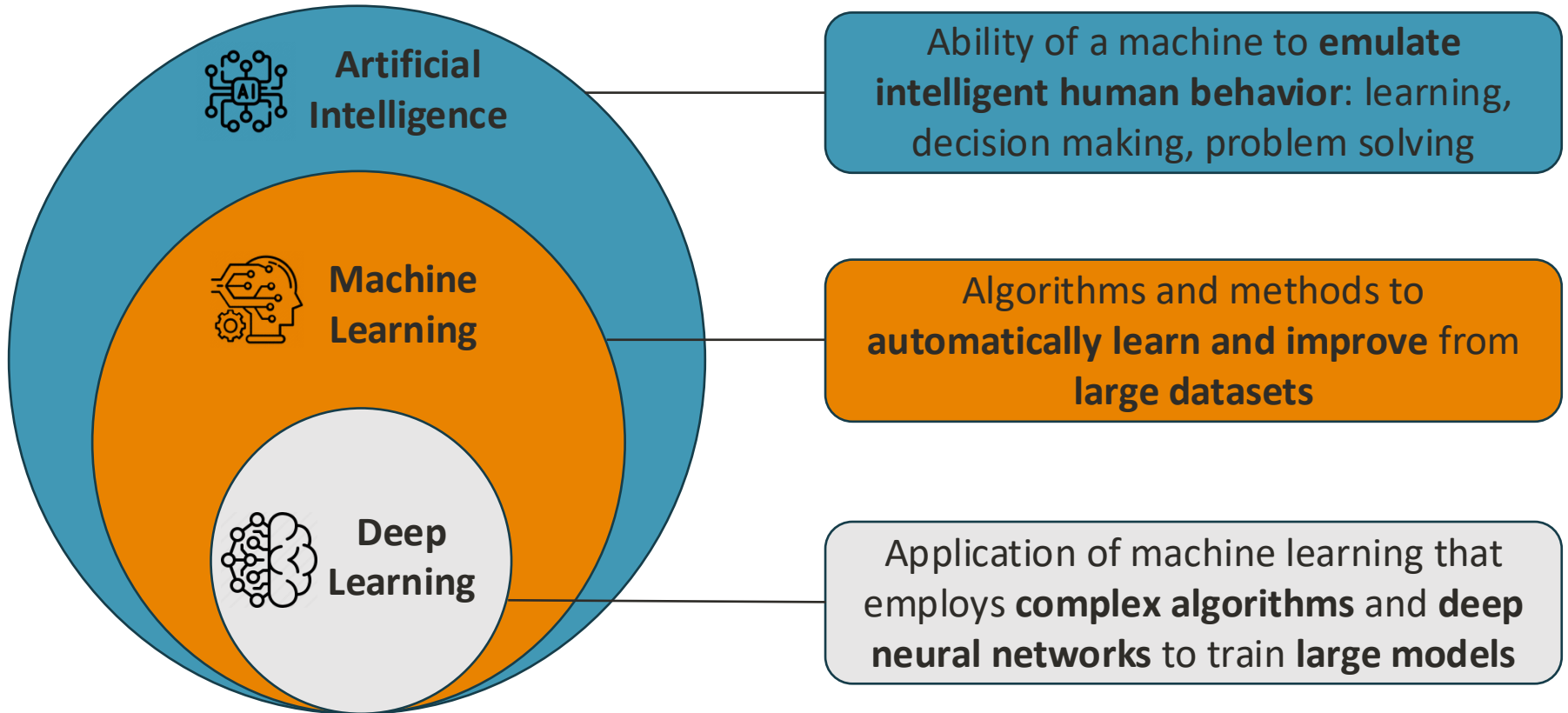
Milestones in ML and AI – tools



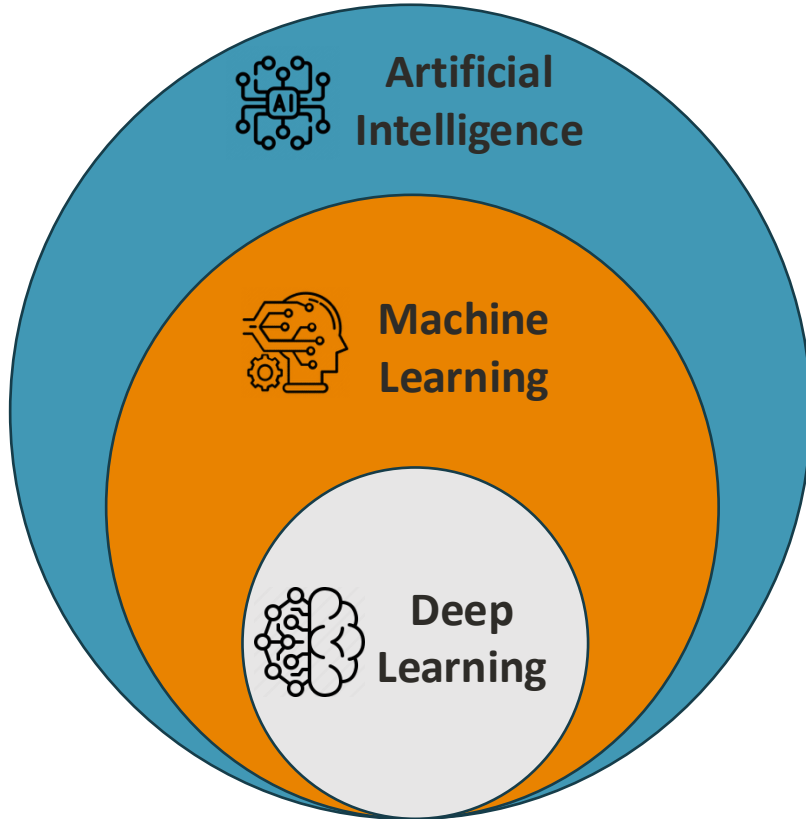
Milestones in ML and AI – foundation models



ML and AI – definitions



ML and AI – scientific process



1. Formulating hypothesis

- Brainstorming
- Retrieving knowledge

2. Planning test

- Identify test conditions
- Develop test environments

3. Conducting tests

- Obtaining data
- Controlling experiments
- Running calculations

4. Distilling knowledge

- Processing data
- Visualizing data
- Discovering patterns

Scientific Process

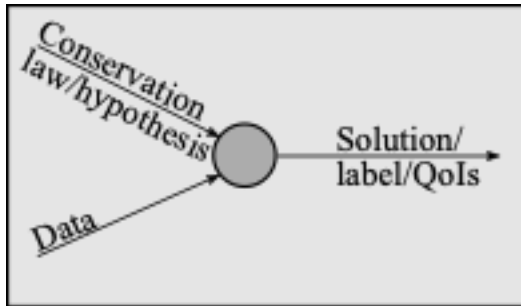
What is Machine Learning?

- “Learning is any process by which a system improves performance from experience” (Herbert Simon)
- Machine Learning is the study of algorithms (Tom Mitchell (1998)) that
 - improve their performance P
 - at some task T
 - with experience E
- Well-defined learning task is given by $\langle P, T, E \rangle$

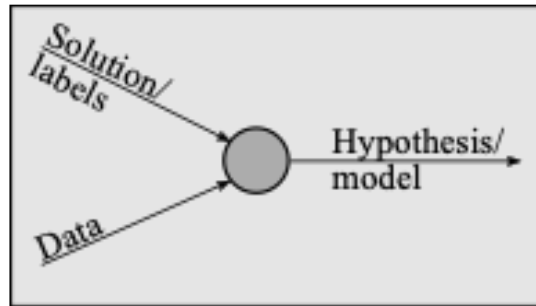
What is Machine Learning?

- Knowledge discovery paradigms

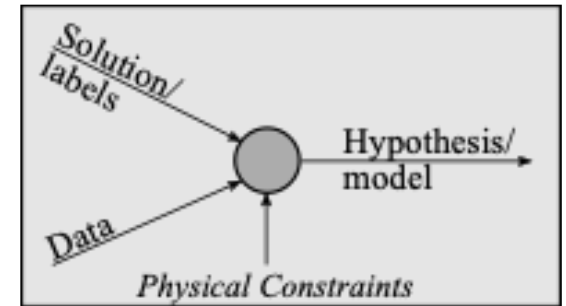
Physics-based
approach



Data-driven
approach



Knowledge-guided data
driven approach



ML application

- Utility of ML/AI
 - Insufficient expertise: new environments, conditions
 - No fundamental understanding and lack of explanation
 - Customizable and adaptive models
 - Models are based on large data
- Combustion-specific ML applications
 - Pattern recognition: combustion instabilities
 - Pattern generation: statistical properties
 - Anomaly recognition: engine failure, unstart, flame blow-out
 - Prediction: accelerate simulations, samples
- But learning is not always useful
 - Comprehensive physical/mathematical foundation
 - Lack of data

Challenges in ML, applied to combustion?

- Poor regression performance and physics principles
 - Black-box modeling and uncertainty quantification
 - Lack of structured and diverse datasets
-
- We will address some of these issues!



Mathematical Background

Mathematical background: Probabilistic analysis

- Probability density function (PDF) and probability

$$p_x(X)dx = \Pr\{X \leq x < X + dX\}$$
$$\iff \Pr\{X_a \leq x < X_b\} = \int_{X_a}^{X_b} p_x(X)dX$$

with

- Random variable: $x \in \{T, p, Y_i, E, H, \dots\}$
 - Sample-space variable: X
- Properties

$$p_x(X) \geq 0 \quad \int_{-\infty}^{\infty} p_x(X)dX = 1$$

Mathematical background: Probabilistic analysis

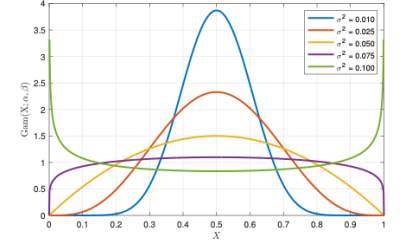
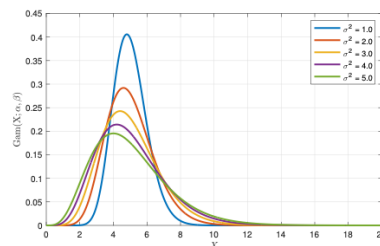
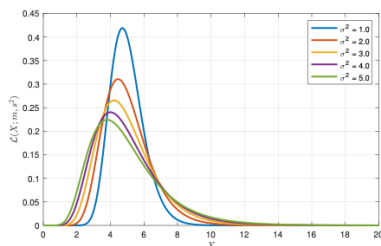
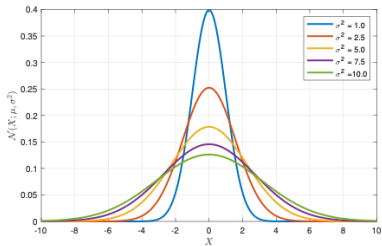
- Presumed pdfs: Approximation of PDFs by analytic functions:

- Gaussian: $p_x(X) = \mathcal{N}(X; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(X - \mu)^2}{2\sigma^2} \right\}$

- Log-normal: $p_x(X) = \mathcal{L}(X; m, s^2) = \frac{1}{\sqrt{2\pi\sigma^2}X} \exp \left\{ -\frac{(\ln(X) - m)^2}{2s^2} \right\}$

- Gamma: $p_x(X) = \text{Gam}(X; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} X^{\alpha-1} \exp \{-\beta X\}$

- Beta: $p_x(X) = \beta(X; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} X^{\alpha-1} (1 - X)^{\beta-1}$



Mathematical background: Probabilistic analysis

- Joint PDF: $p_{x,y}(X, Y)$

- Marginal PDF: $p_x(X) = \int_{-\infty}^{\infty} p_{x,y}(X, Y) dY$

- Conditional PDF

$$p_{x,y}(X, Y) = p_{x|y}(X|Y)p_y(Y) \iff p_{y|x}(Y|X) = \frac{p_{x|y}(X|Y)p_y(Y)}{p_x(X)}$$

(Bayes' theorem)

- Bayesian inference

$$p_{\theta|\ell}(\Theta|\mathcal{L}) = \frac{p_{\ell|\theta}(\mathcal{L}|\Theta)p_{\theta}(\Theta)}{p_{\ell}(\mathcal{L})}$$

Parameter Data

Likelihood function

Prior probability

Marginal likelihood function

Mathematical background: Probabilistic analysis

- Consider data as discrete random variables: $\{x_1, x_2, \dots, x_N\} = \{x_i\}_{i=1}^N$

- Fine-grain distribution

$$p_x^*(X) = \frac{1}{N} \sum_{i=1}^N \delta(X - x_i)$$

- Probability mass function (PMF)

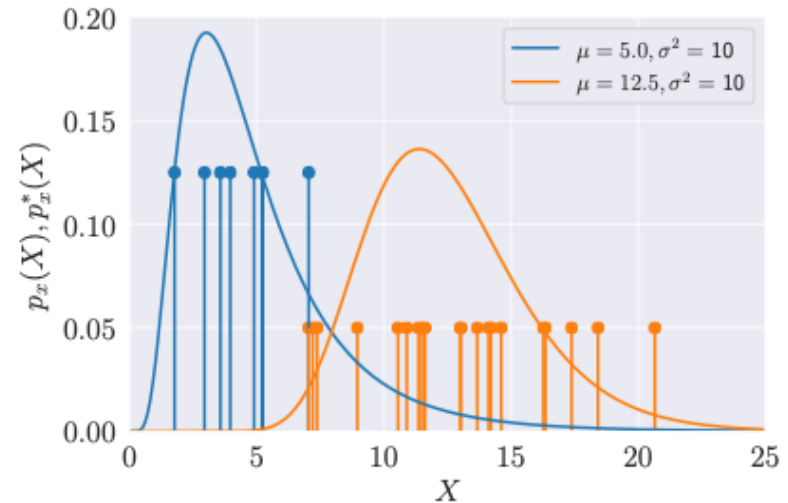
$$P_x(X_i) = \frac{1}{N} \mathbb{I}(X - x_i)$$

- Properties:

$$0 \leq P_x(X_i) \leq 1 \quad \forall X_i$$

$$\sum_{i=1}^N P_x(X_i) = 1$$

$$\begin{aligned} \int_{-\infty}^{\infty} p_x^*(X) dX &= \frac{1}{N} \sum_{i=1}^N \int_{-\infty}^{\infty} \mathbb{I}(X) \delta(X - x_i) dX, \\ &= \frac{1}{N} \sum_{i=1}^N \mathbb{I}(X = x_i), \\ &= \sum_{i=1}^N P_x(X_i), \end{aligned}$$



Mathematical background: Loss function

- Statistical model representation

$$(\mathcal{L}, \{P_{x|\theta}\}_{\theta \in \mathcal{P}})$$

Sample space: $\mathcal{L} = \{(x, y)_1, (x, y)_2, \dots, (x, y)_N\} = \{(x, y)_i\}_{i=1}^N$

Parameterized family of PDFs defined on \mathcal{L}

- Example: Extinction/ignition event with binary outcome
 - Sample space: $\mathcal{L} \in \{0, 1\}$
 - Distribution (Bernoulli): $P_{x|\theta}(X|\theta) = \theta^X (1 - \theta)^{1-X}$

Mathematical background: Loss function

- Loss functions
 - Performance measurement: metric to evaluate model performance by quantifying difference between predictions and actual results
 - Direction for improvement: guide improvement by directing algorithm to adjust parameters to reduce loss and improve predictions
 - Balancing bias and variance: Effective loss functions help balance model bias (oversimplification) and variance (overfitting), essential for model generalization
 - Influencing model behavior: loss functions can affect model behavior: improve robustness, prefer specific types of errors)

Mathematical background: Loss function

- Choice of loss function

Factor	Description
Type of Learning Problem	Classification/Regression, Binary vs. Multiclass Classification
Model Sensitivity to Outliers	Some loss functions are more sensitive to outliers (MSE), while others are more robust (MAE)
Desired Model Behavior	Influences how model behaves
Computational Efficiency	Computational intensity of evaluating loss functions
Convergence Properties	Smoothness and convexity of a loss function can affect the ease and speed of training
Scale of task	For large-scale tasks, a loss function that scales well and can be efficiently optimized is crucial

Mathematical background: Loss function

- Loss functions

- Mean absolute error: $E_{MAE} = \frac{1}{N} \sum_{i=1}^N \|y_i - f(x_i)\|$

- Linear error

- Minimize outlier impact

$$E_{MSE} = \frac{1}{N} \sum_{i=1}^N \|y_i - f(x_i)\|^2$$

- Mean square error:

- Amplify outlier impact

- Binary cross-entropy loss: $E_{BCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(f(x_i)) + (1 - y_i) \log(1 - f(x_i))]$

- Penalization for confident yet incorrect predictions

- Other loss functions

- Kullback-Leibler (KL) Divergence; Hinge loss, Huber loss;

Mathematical background: Loss functions

- Model evaluation metrics: assess model performance

- Regression: MSE, MAE, R^2

- Classification Metrics

- Accuracy: $A = \frac{TP+TN}{TP+TN+FP+FN}$

- Only gives info on true predictions, not on mistakes

- Precision: $P = \frac{TP}{TP+FP}$

- Lets you know when you have false positives (bank fraud)

- Recall: $R = \frac{TP}{TP+FN}$

- Lets you know when you have false negatives (COVID)

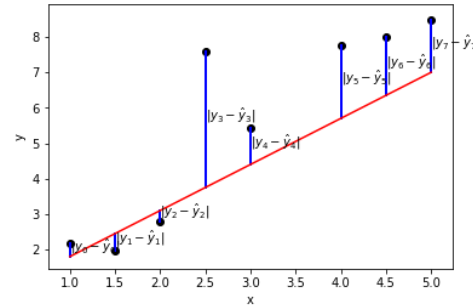
- F1-score: $F1 = \frac{2P R}{P+R}$

- Gives both FP and FN (search applications)

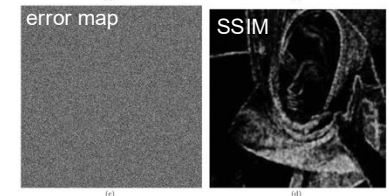
- ROC, AUC

- Computer vision

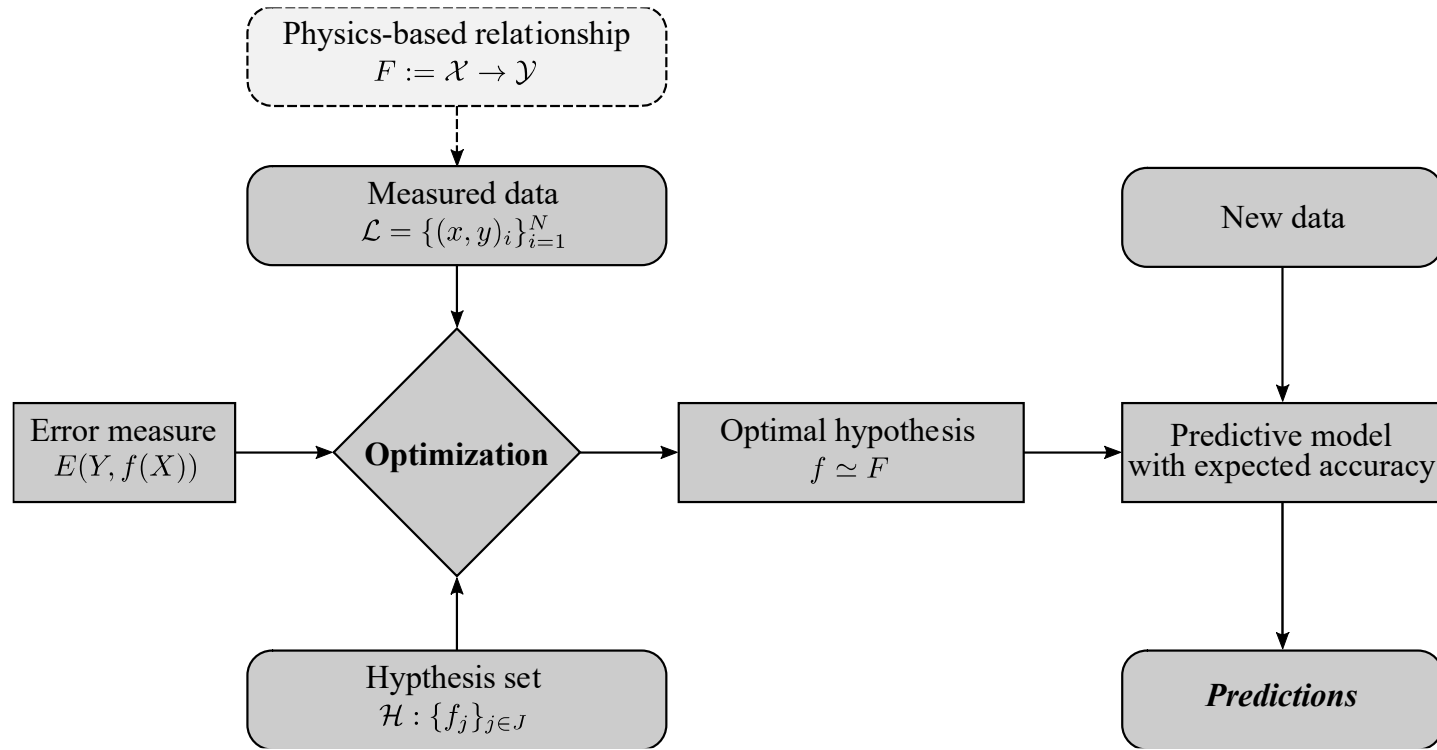
- structural similarity index: $SSIM = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}$



		Predicted Value	
		Positive	Negative
Actual Value	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

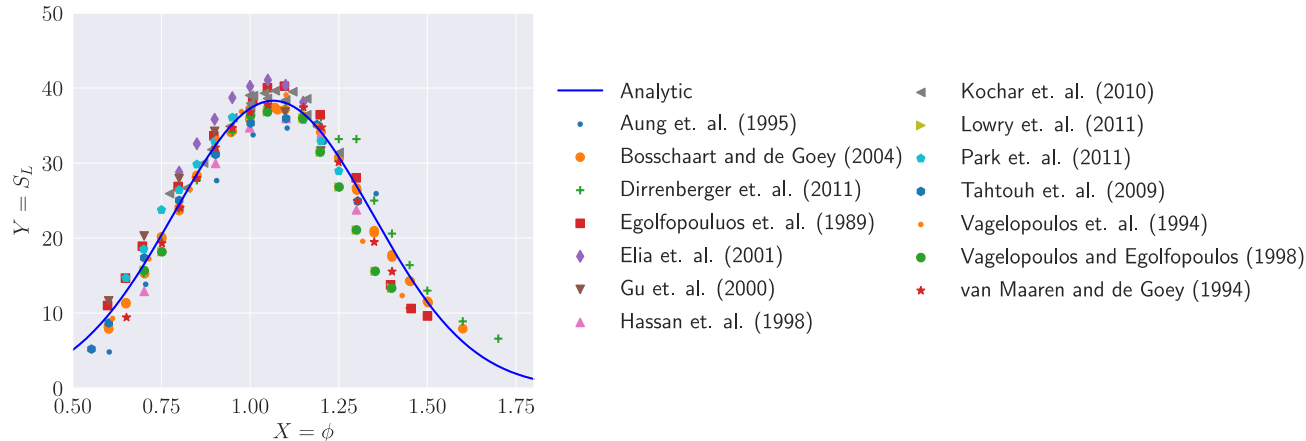


A generic supervised learning algorithm



Learning models: regression

- Learning problem: polynomial approximation of flame-speed for CH₄/Air mixture at ambient conditions

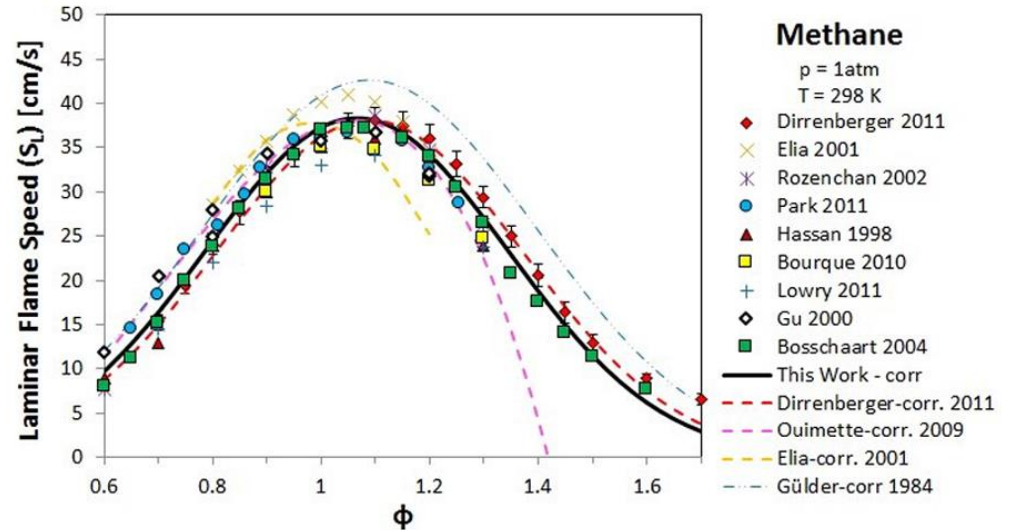


Learning models: regression

- Learning problem: regression of flame laminar speed s_L for CH₄/Air mixture at ambient conditions

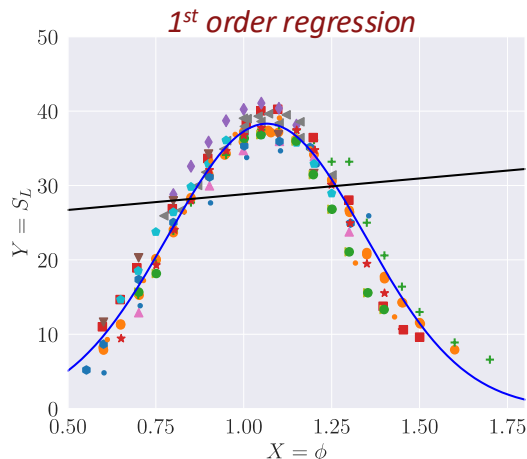
- Empirical correlation: $s_L^0(\phi) = W\phi^\eta \exp\{-\xi(\phi - \sigma)^2\}$
with:

- $W = 42.2 \frac{\text{cm}}{\text{s}}$
- $\eta = 0.15$
- $\xi = 5.18$
- $\sigma = 1.075$

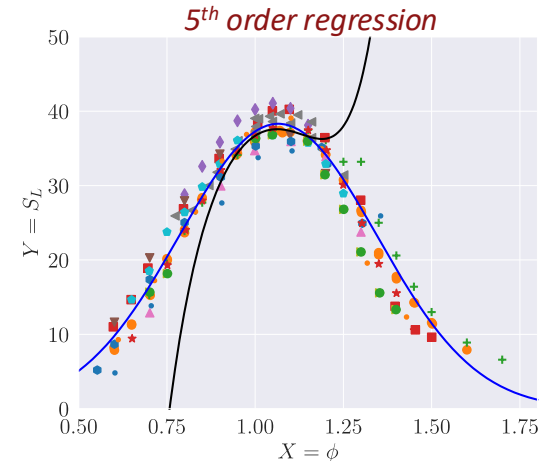
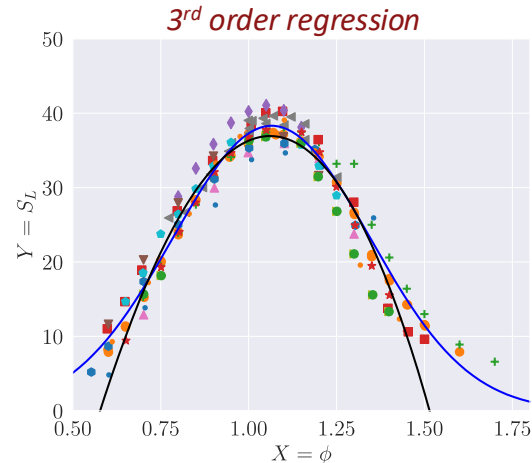


Learning models: regression

- Learning problem: regression of flame laminar speed s_L for CH₄/Air mixture at ambient conditions
- Polynomial approximation of flame-speed s_L for CH₄/Air mixture at ambient conditions: $y = \sum_{i=0}^N a_i x$



Underfitting: low variance, high bias



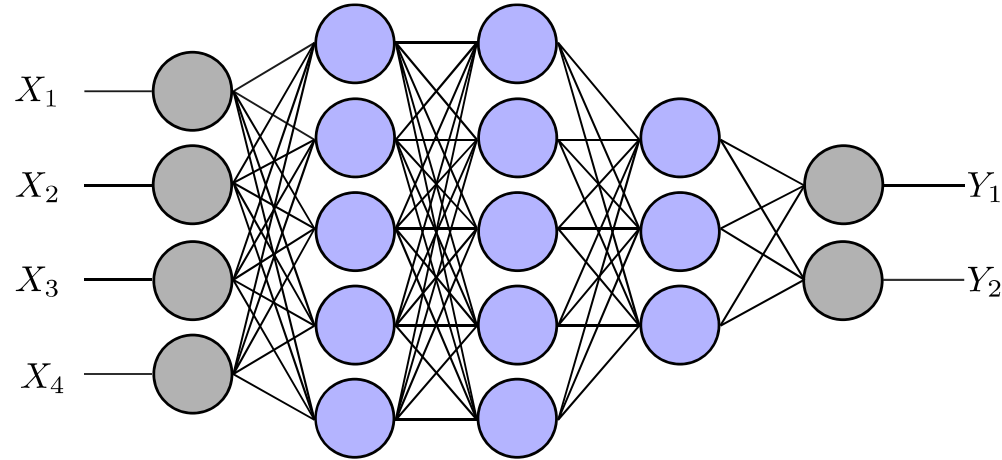
Overfitting: high variance, low bias

Learning models: regression

- Learning problem: regression of flame laminar speed s_L for CH4/Air mixture at ambient conditions
- Polynomial approximation of flame-speed s_L for CH4/Air mixture at ambient conditions: $y = \sum_{i=0}^N a_i x$
- Generalization error:

$$\mathbf{E}\{\mathcal{E}(f)\} = \underbrace{E(Y, f_B(X))}_{\text{Noise}} + \underbrace{(f_B(X) - \mathbf{E}\{f(X)\})^2}_{\text{Bias}} + \underbrace{\mathbf{E} \left[(\mathbf{E}\{f(X)\} - f(X))^2 \right]}_{\text{Variance}}$$

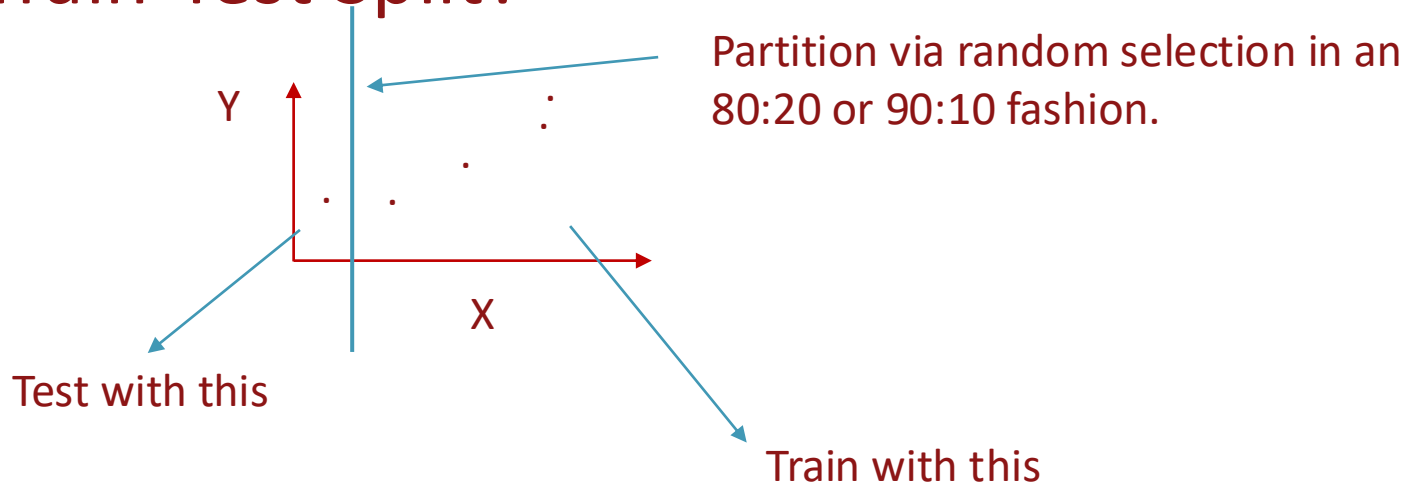
Hyperparameter search



How many neurons in each layer?
How many layers?
What optimization scheme?
How much data?
What forward operation?
What non-linearity?
NN or other method?

1. Intuition/Domain Knowledge
2. Exhaustive Search (Expensive!)
3. Random Search (Not efficient)
4. Bayesian Optimization (Gaussian Assumptions)
5. Reinforcement Learning
6. AutoML
7. ...

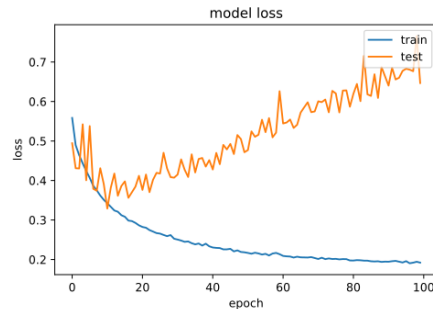
Why do a Train-Test Split?



Why? -> Can Diagnose Overfitting

Loss is your objective function

- Likelihood
- Error between points

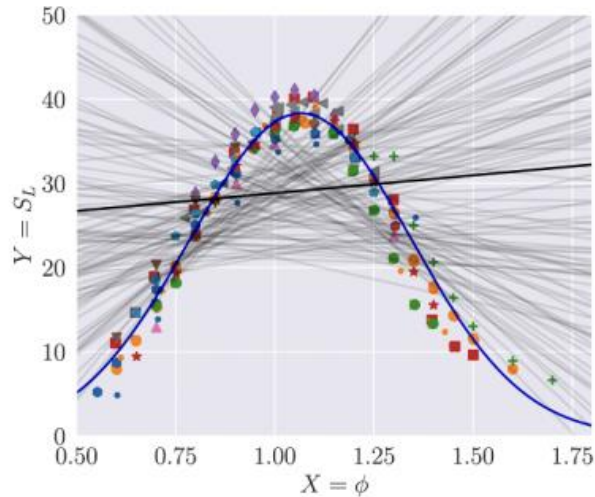


This can happen if your hyperparameter search arrives at a model that only suits your train data

Why does overfitting occur?

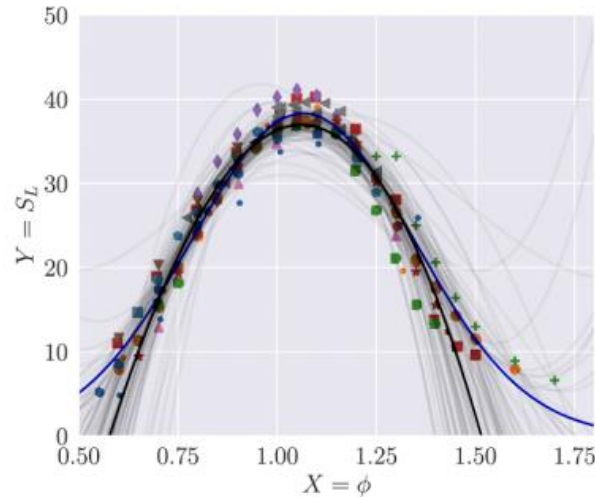
- We train 100 linear regression models of different complexity
- More complex models (more layers, etc) can arrive at high bias.

Underfit
High variance



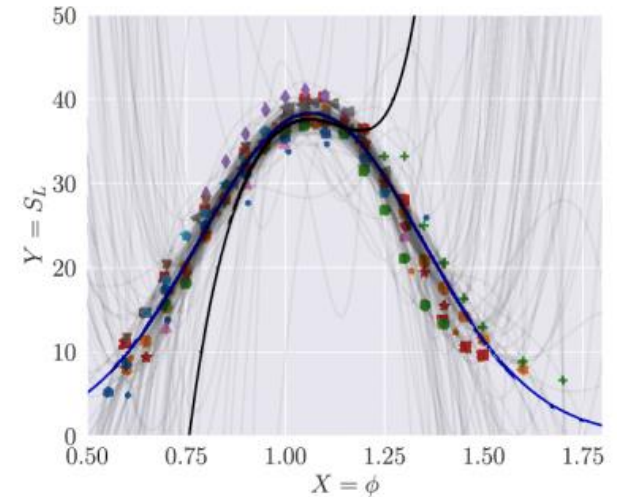
(a) 1st order regression.

Decent fit



(b) 3rd order regression.

Overfit
High-bias



(c) 5th order regression.

Better split: Train-test-validation split

- Train-test split let's you **diagnose overfitting!**
 - Doesn't actually do anything to stop this
- Train-val-test split can help alleviate overfitting
 - Hyper-parameter search on a validation set that is different from train and test set
- Partition via random selection in an 80:10:10 or 90:5:5 fashion.

Overfitting can happen
if your hyperparameter
search arrives at a
model that that only
suits your train data

Other ways to deal with overfitting

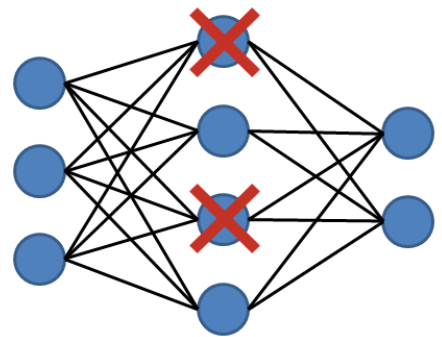
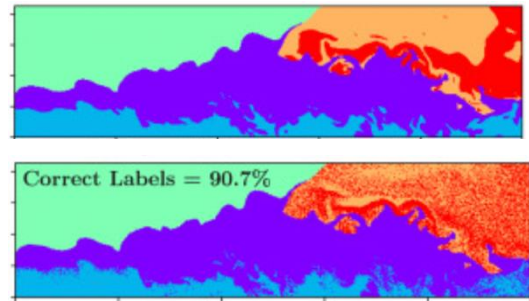
- Regularization
- Data augmentation
- Dropout
- Batch Normalization

L1 Regularization

$$\text{Cost} = \underbrace{\sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij}W_j)^2}_{\text{Loss function}} + \lambda \underbrace{\sum_{j=0}^M |W_j|}_{\text{Regularization Term}}$$

L2 Regularization

$$\text{Cost} = \underbrace{\sum_{i=0}^N (y_i - \sum_{j=0}^M x_{ij}W_j)^2}_{\text{Loss function}} + \lambda \underbrace{\sum_{j=0}^M W_j^2}_{\text{Regularization Term}}$$



Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
 Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma,\beta}(x_i) \quad // \text{ scale and shift}$$

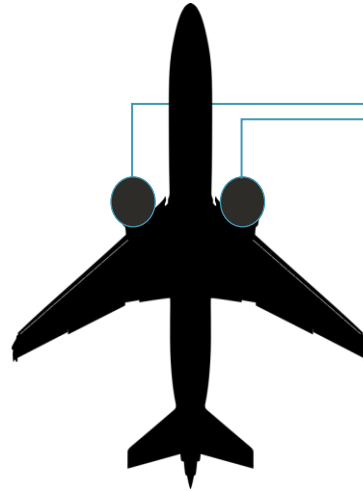
Question

- Why is regression different from ML?
- How can we assess model convergence?
- How to introduce uncertainties?

Module 2: Data and Data Curation

Data in combustion science and engineering

- Sensors and actuators
 - System prognostics
 - Health monitoring
 - Stall and flow control
 - Clearance
 - Emissions
 - Cooling



$$20 \text{ TB} \times 2 \times 6 \times 20,000$$

Data per engine
and hour

Twin-engine
Boeing 737

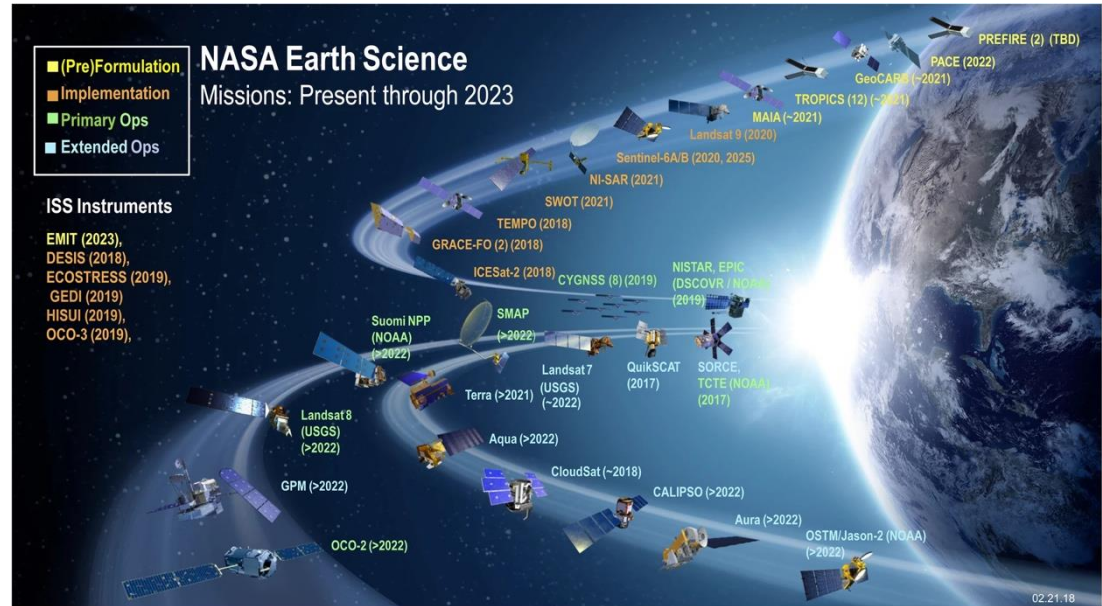
6h flight:
JFK → LAX

USA daily
commercial flights

$$= 4.8 \text{ EB/day}$$

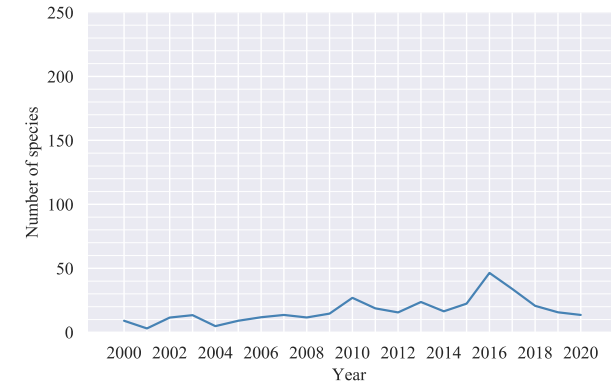
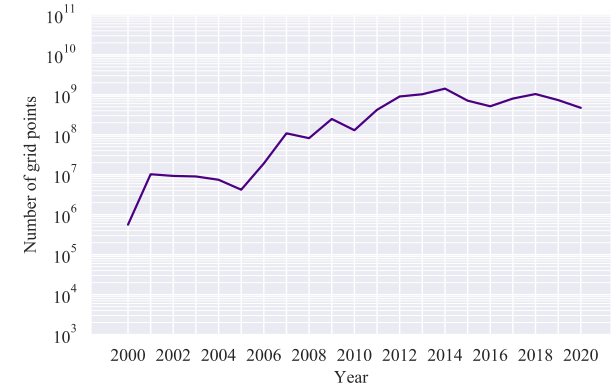
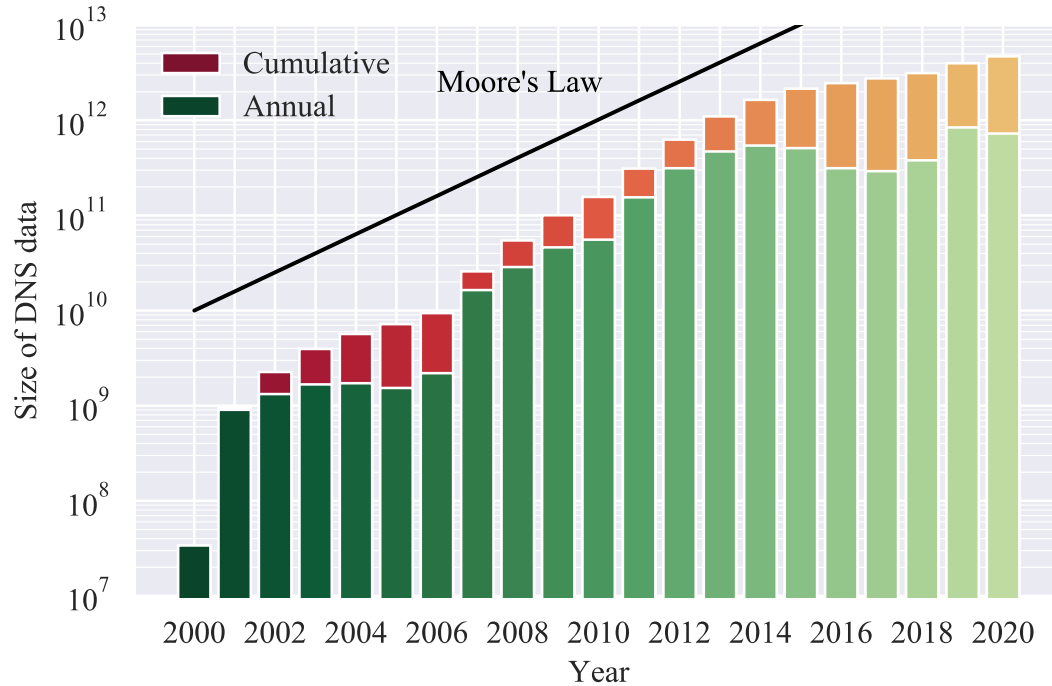
Data in combustion science and engineering

- Satellite observations
 - Wildfires
 - Vegetation, fuel density
 - Moisture, humidity
 - Emissions, pollutants
 - Smoke



> 1TB/day/satellite

Data in combustion science and engineering



Ihme, Chung, and Mishra, Prog. Energy Combust. Sci. 91, 101010, 2022

What makes a good dataset?

SCALE & DIVERSITY!!

SUN, 131K

[Xiao et al. '10]

LabelMe, 37K

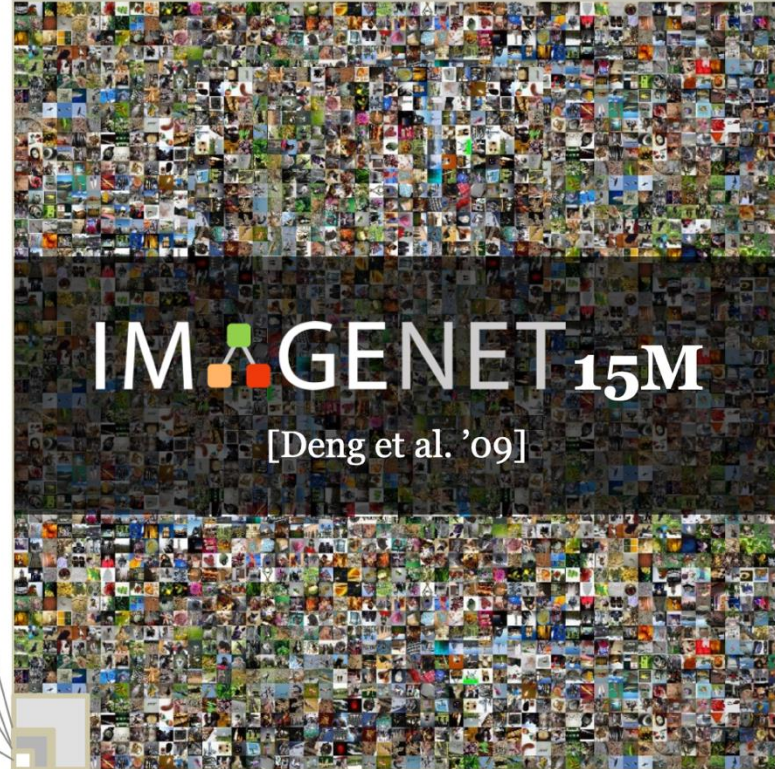
[Russell et al. '07]

PASCAL VOC, 30K

[Everingham et al. '06-'12]

Caltech101, 9K

[Fei-Fei, Fergus, Perona, '03]

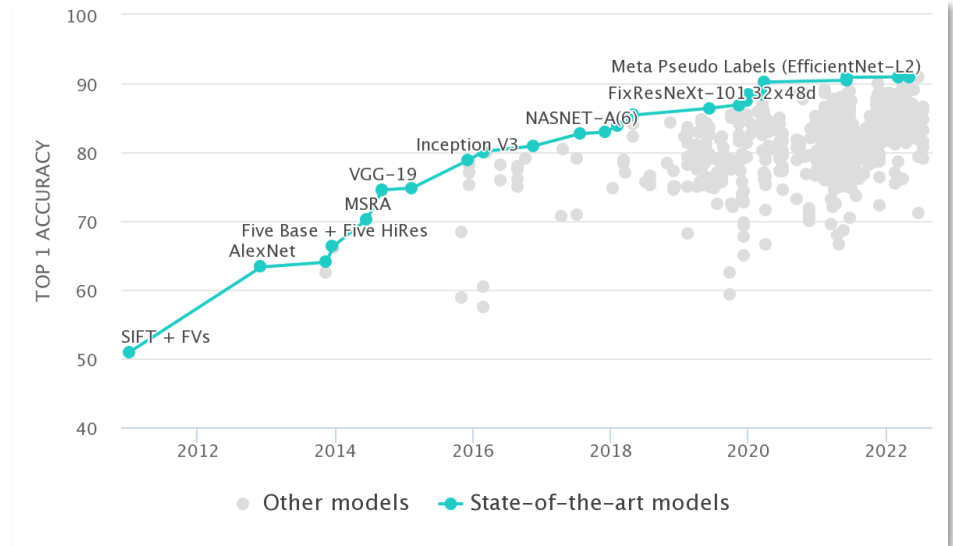


Challenge: Data for combustion

ImageNet Computer Vision Database



- 2D image format
- 15M samples in total (~1 TB)
- Diversity: 1000 objects



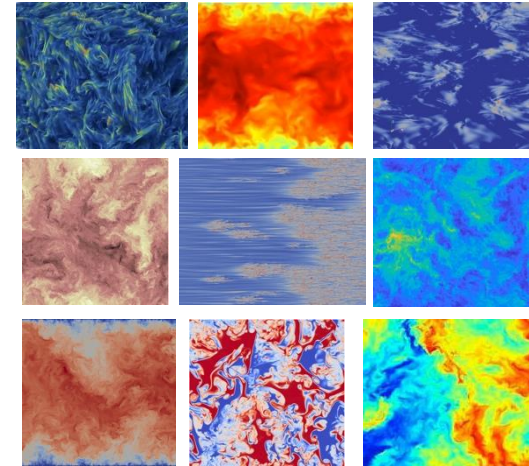
Challenge: Data in Flow Physics

ImageNet Computer Vision Database



- 2D
- 15M samples in total (~1 TB)
- Diverse - 1000 objects
- 10^6 bytes/image

Johns Hopkins Turbulence Database



VS

- **3D + Time**
- 16K samples in total (**500 TB**)
- **9 configurations**
- **10^{11} bytes/snapshot**

How to share large datasets?

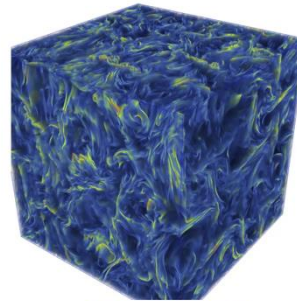
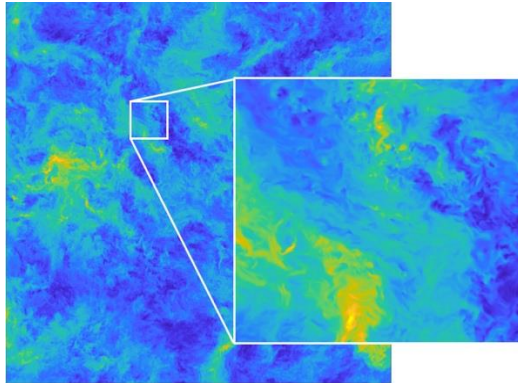
Li Fei-Fei and Jia Deng. (2017). "ImageNet: Where have we been? Where are we going?" In: CVPR 2017 Workshop.

Yi Li et al. (2008). "A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence." In: Journal of Turbulence 9:N31.

State-of-the-art Flow Physics Scientific Dataset

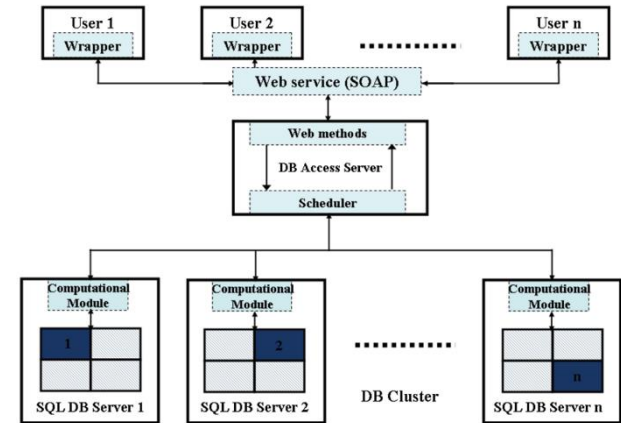
- Johns Hopkins Turbulence Database
- Only 9 3D configurations
- Total Size = 500 TB (3D, Detailed)
- State of the art snapshots > 100 GB

Extremely Detailed!



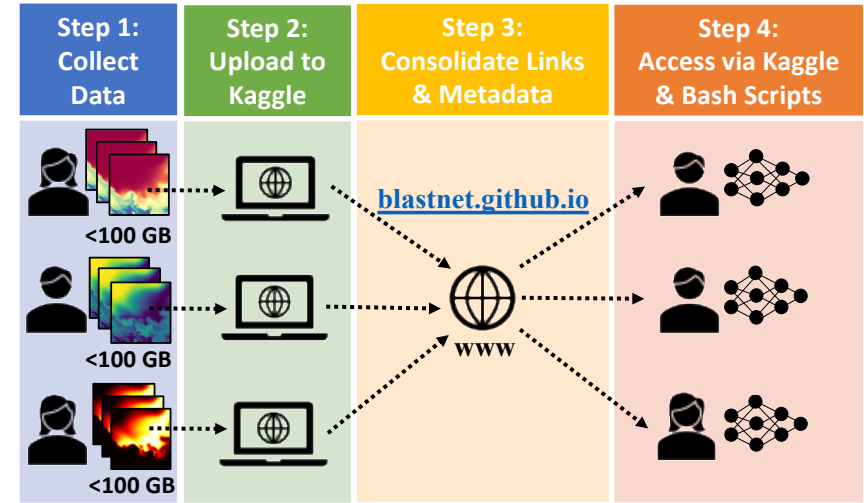
Current field $\nabla \times \mathbf{B}$

Hosted on dedicated infrastructure



Large Public Combustion Dataset through Community

- BLASTNet (Bearable Large Accessible Scientific Training Network-of-Datasets) as scalable data access through community involvement
 - 25 TB of DNS data
 - 36 configurations
 - Consistent data format
 - 100 ML models
 - Code, tutorials to browse data, ML training
- Public Repositories (Kaggle) provide 200 GB free storage per repository

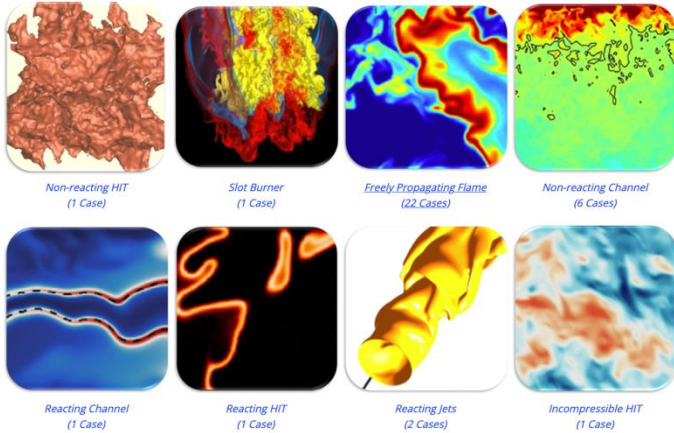


Data contributed by combustion/fluids community:

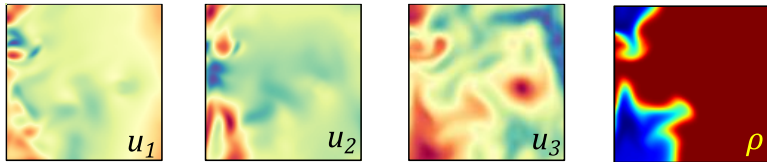


<https://blastnet.github.io>

Measuring Diversity of BLASTNet



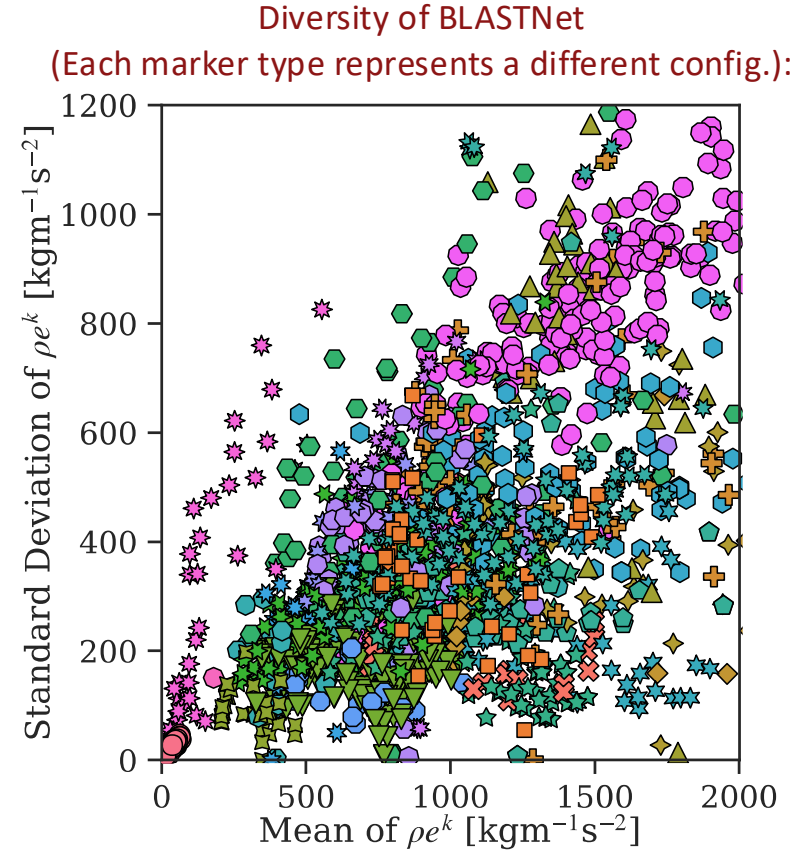
Extract



Collect statistics from each 128^3 sample

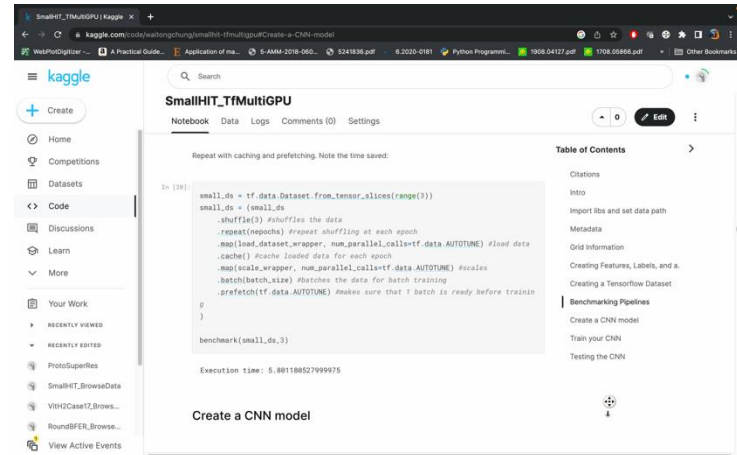
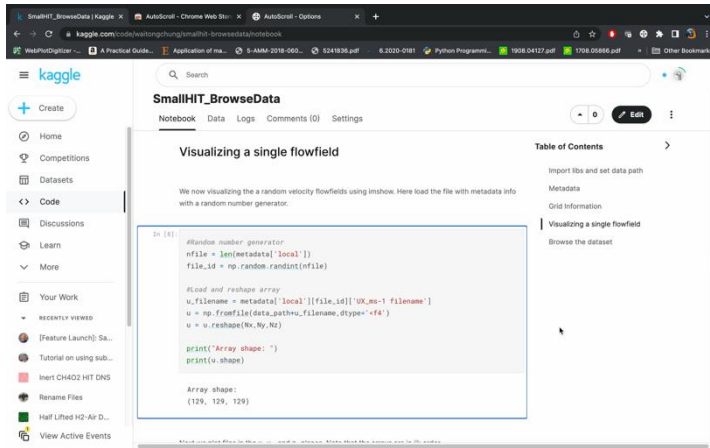


- 128^3 3D sub-volumes of velocity and density.



BLASTNet: Tutorials

- You can browse and run code on Kaggle cloud computing infrastructure
- Tutorials for browsing data, and multi-GPU training is available

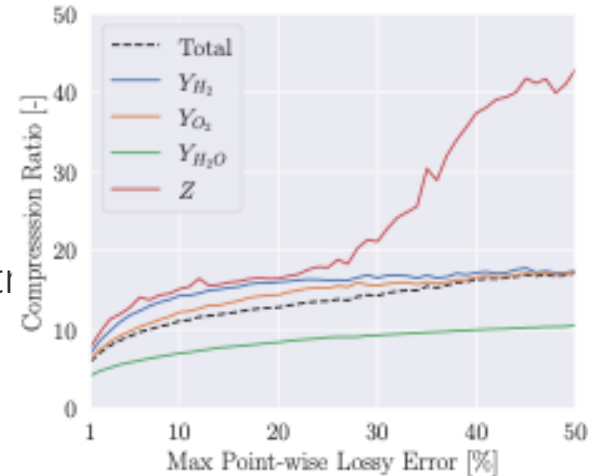


How to further boost scale?

- Overcoming Kaggle's 100 GB per upload limit.



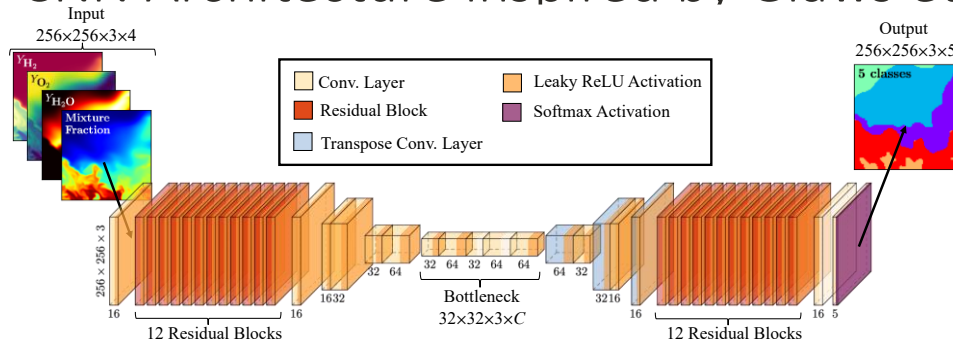
- Compression performed via SZ2 [1].
 - Partitions domain into different clusters
 - Searches for candidate regression functions
 - Stores data as quantized regression coefficients are matrix
 - Enables **point-wise error control**.



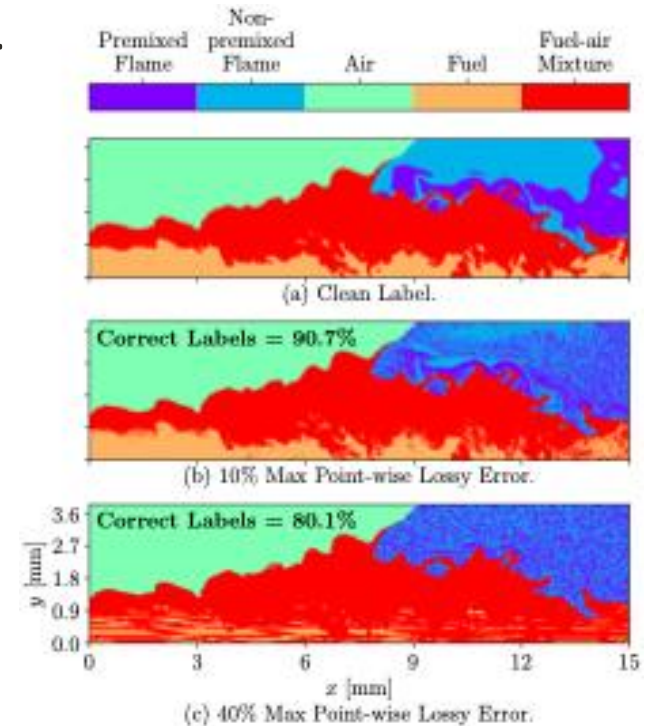
[1] Xi, Liang, S. Di, D. Tao, S. Li, Li, H. Guo, Z. Chen, and F. Cappello. *IEEE Big Data* (2018).

Current Study: How does Lossy data affect ML training?

- Test on a semantic segmentation problem with 4 features
- 3-D CNN Architecture inspired by Glaws et al.



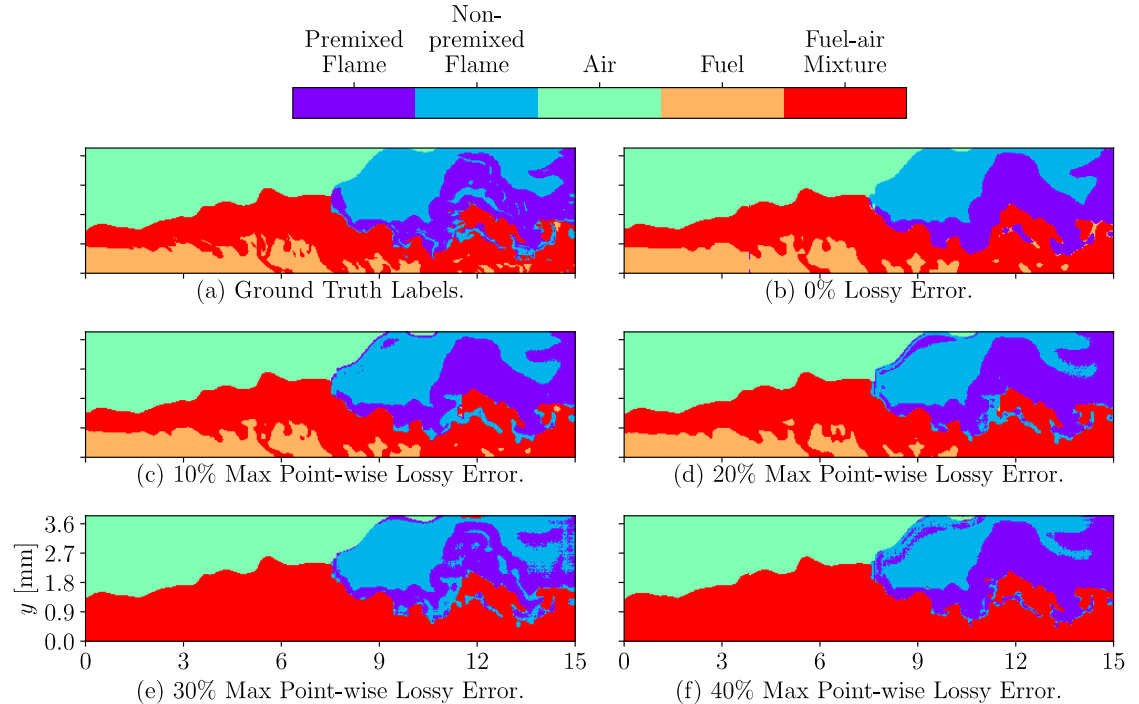
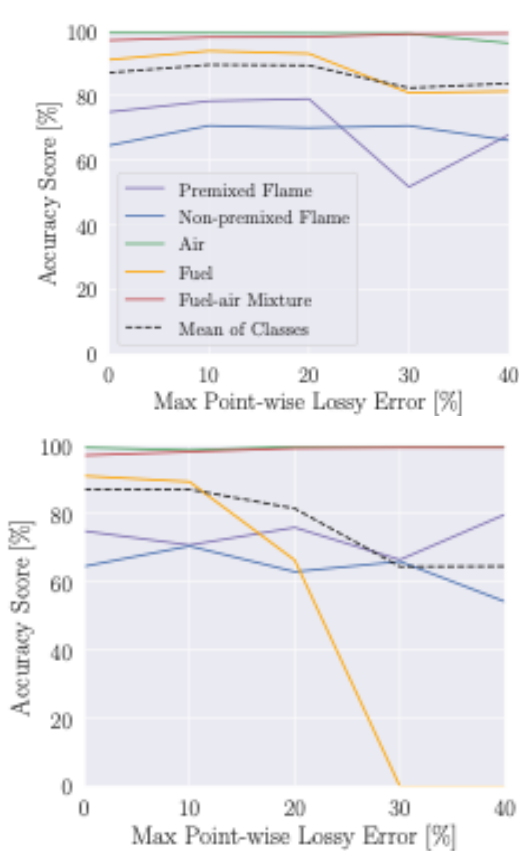
- Data from 1.28 B cell H_2 reacting flow simulation from Jung et. al.
- Apply different levels of lossy compression on this data then investigate ML accuracy



A. Glaws, R. King, and M. Sprague. *Phys. Rev. Fluids* (2020).

K. S. Jung, S. O. Kim, T. Lu, J. H. Chen, and C.S. Yoo. *Combust. Flame* (2021)

Current Study: How does Lossy data affect ML training?



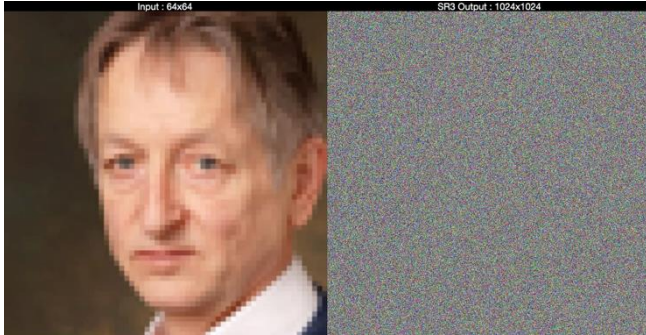
Deep learning is robust to 10% lossy error/noise!

What can be done with BLASTNet?

Benchmarking and Scaling

Super-Resolution via Deep Learning

Google SR3 Super-Resolution Model

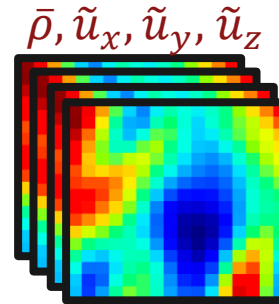


Improving Coarse Simulations

Filtered Momentum Equation

$$\partial_t(\bar{\rho}\tilde{\mathbf{u}}) + \nabla \cdot (\bar{\rho}\tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}}) = -\nabla\bar{p} + \nabla \cdot (\bar{\boldsymbol{\tau}} + \boldsymbol{\tau}^{\text{SGS}}), \quad \text{where}$$

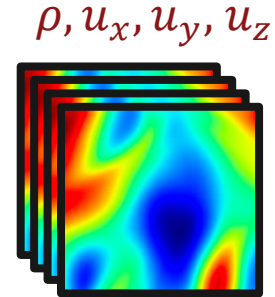
$$\boldsymbol{\tau}^{\text{SGS}} = \bar{\rho}(\widetilde{\mathbf{u} \otimes \mathbf{u}} - \tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}}).$$



Compressible Navier-Stokes

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0,$$

$$\partial_t(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau}$$



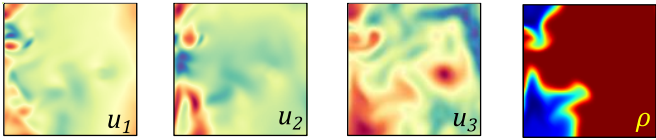
Super-Resolution via Deep Learning

BLASTNet Dataset



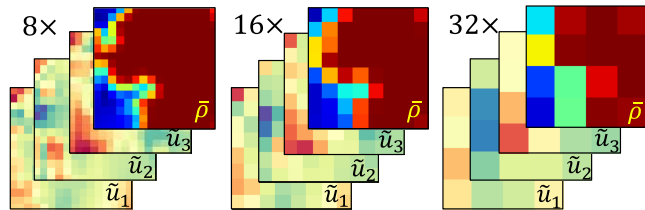
Momentum128 3D SR Dataset

Labels



- **128³ 3D sub-volumes** of velocity and density.
- **27 configs, 67 GB, 2K samples**, 4 channels, uniform grid.

Features



- Filter and downsample labels to generate coarse simulation samples.

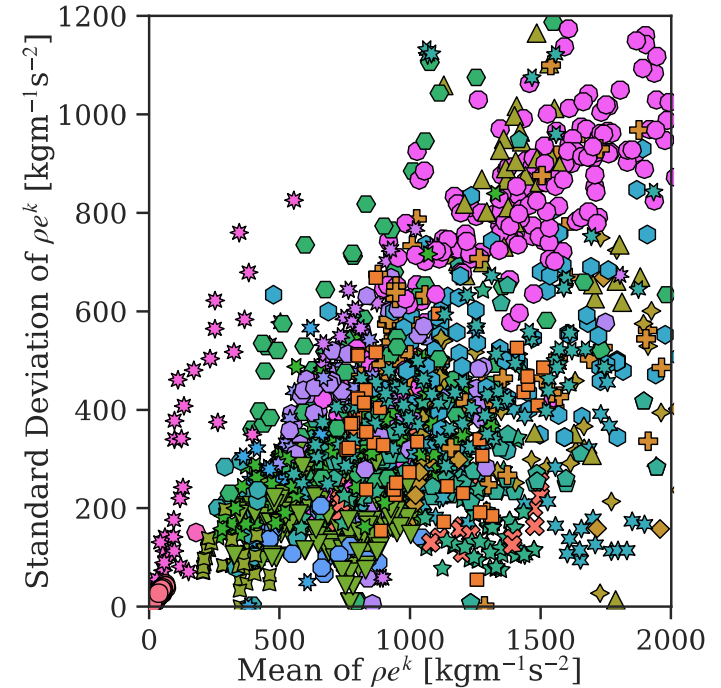
Collect statistics
from each sample



$$\rho e^k = \rho(u_1^2 + u_2^2 + u_3^2)/2$$

Diversity of BLASTNet

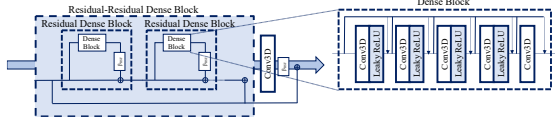
(Each marker type represents a different config.):



Benchmarking 3D Super-Resolution

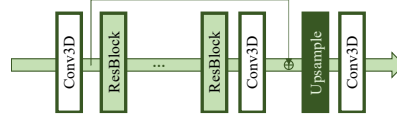
- Trained models via Supervised Learning by Optimizing MSE.

Residual-in-Residual Dense Block

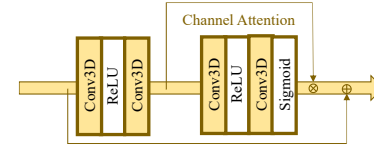


with/without a physics-based loss regularization term.

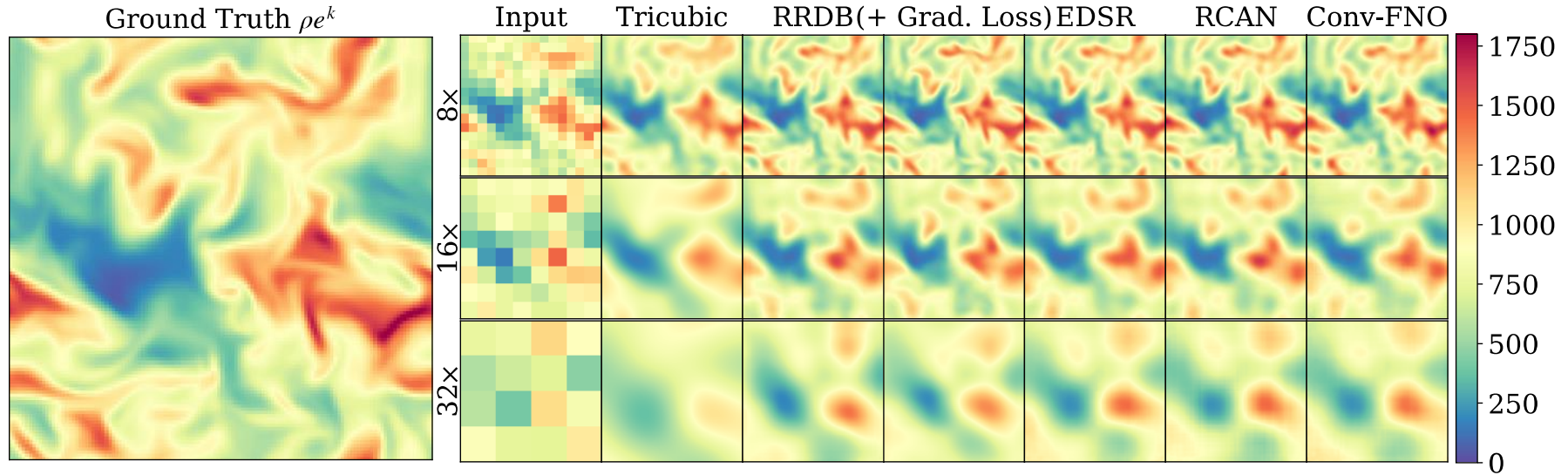
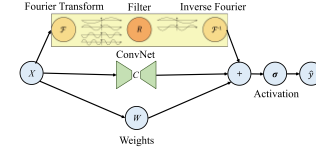
Enhanced Deep Residual Network



Residual Channel Attention Network



Conv. Fourier Neural Operator



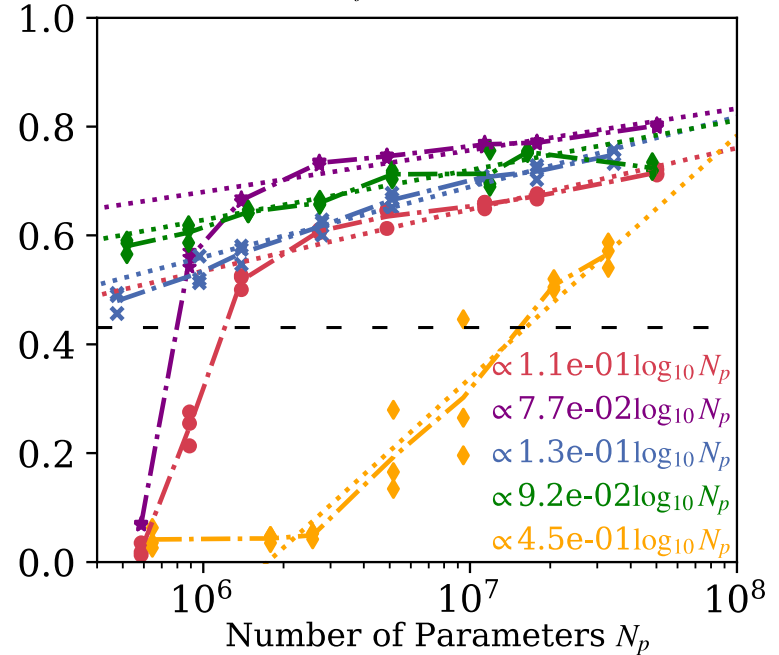
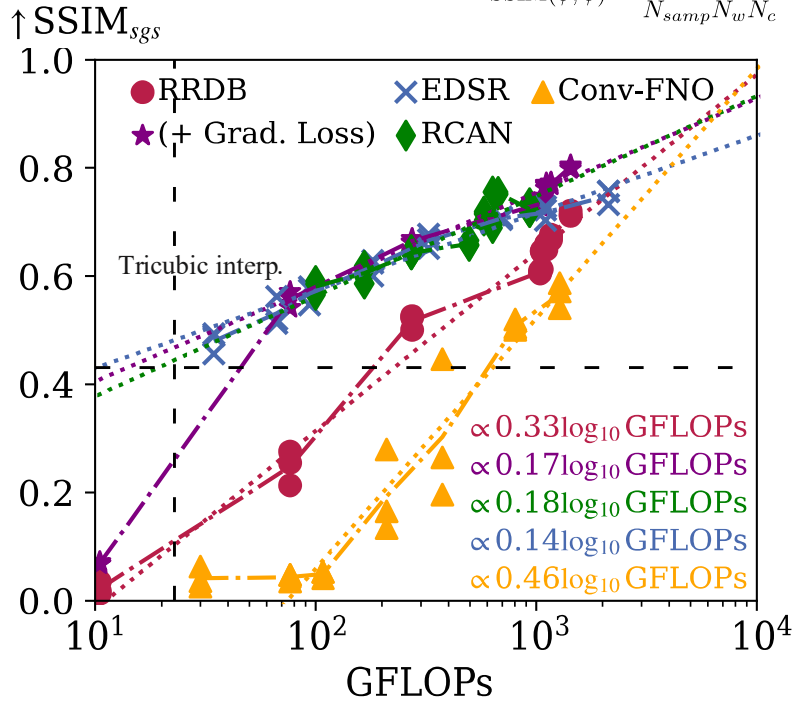
Chung et al. "Turbulence in Focus: Benchmarking Scaling Behavior of 3D Volumetric Super-Resolution with BLASTNet 2.0 Data" NeurIPS 2023

Neural Scaling Laws - 8x Super Resolution


Measures Similarity of Super-resolved Turbulence:

$$\nabla \cdot \tau^{\text{sgs}}$$

$$\text{SSIM}(\hat{\phi}, \phi) = \frac{1}{N_{\text{samp}} N_w N_c} \sum_{i=1}^{N_{\text{samp}}} \sum_{j=1}^{N_w} \left(\frac{2\mu_{\hat{\phi}}\mu_{\phi} + c_1^2}{\mu_{\hat{\phi}}^2 + \mu_{\phi}^2 + c_1^2} \cdot \frac{2\sigma_{\hat{\phi}}\sigma_{\phi} + c_2^2}{\sigma_{\hat{\phi}}^2 + \sigma_{\phi}^2 + c_2^2} \right)_{ij}$$

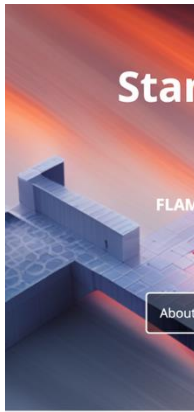


W. T. Chung, B. Akoush, P. Sharma, A. Tamkin, K. S. Jung, J. H. Chen, J. Guo, D. Brouzet, M. Talei, B. Savard, A. Y. Poludnenko, M. Ihme (2023). Turbulence in Focus: Benchmarking Scaling Behavior of 3D Volumetric Super-Resolution with BLASTNet 2.0 Data, Advances in Neural Information Processing Systems 36 (NeurIPS 2023)



What can be done with BLASTNet?

Community building!



Stanford FLAME AI Workshop 2024

FLAME: Future Learning Approaches for Modeling and Engineering

Dates: Sept. 23 to Oct. 18 2024 | Venue: Virtual

- About
- Register
- Flyer
- Organizers
- Email
- Flame AI 2023

Mission

To foster a dynamic forum for exchanging ideas, cutting-edge data, advanced methods, and models related to ML techniques for fluid dynamics, turbulence, and environmental science - applications crucial to the development of sustainable energy solutions, climate resilience, wildfire mitigation, and safety systems.

Agenda

1. A **10-day-long** modeling in teams will be held to tackle ML challenges in predicting spatio-temporal and dynamical processes in physical sciences, environmental flows, and engineering with open-source data.
2. **Daily talks** guests, and

Fundamentals



AI for Science



Mission

To foster a dynamic forum for exchanging ideas, cutting-edge data, advanced methods, and models related to ML techniques for fluid dynamics, turbulence, and environmental science - applications crucial to the development of sustainable energy solutions, climate resilience, wildfire mitigation, and safety systems.

Agenda

1. A **four-week long ML challenge** (involving 1-2 person(s) per team) will be held to tackle ML challenges in predicting spatio-temporal and dynamical processes in physical sciences, environmental flows, and engineering with open-source data.
 - o **Prize:** Total of \$5000 for winning teams of ML challenge and an invitation for leading contributors to a joint publication.
2. **Invite talks and keynote lectures** on cutting-edge trends will be given by AI/ML experts within Stanford, academic guests, and industry partners from the Greater Silicon Valley ecosystem. Topics include:

Invited Speakers (In alphabetical order)



Arima Anandkumar
Professor, Google
Sr. Director of AI Research, Nvidia
"AI for Accelerating Scientific Discovery with Simulations and Design"



Steve Brunton
Professor, University of Washington
"Machine Learning for Scientific Discovery with Applications in Fluid Mechanics"



Alex Cramer
Assistant Professor, University of Cambridge
"The Most Great Scientific Theory is Hiding Inside Your Neural Network"



Travis Cizmas
Professor, North Carolina State University
"Work Experiments in Model: Challenges and New Opportunities for Fuelled Combustion"



Stephan Hoyer
Google Research
"Deep Learning with Differentiable Physics for Fluid Dynamics and Weather Forecasting"



George Karniadaki
Professor, Brown University
"Physics-informed Neural Networks (PINNs) and Neural Operators for Fluid Mechanics and Reactor Transport"



Pablo Neumann-Holts
Professor, Harvard University
"Scientific Computing and Machine Learning: There is a Place of Room in the Middle"



Aaron Lee
PhD Student, Stanford University
"An Introduction to Score-Based Diffusion Models"



Alessandro Parente
Professor, University of Illinois at Urbana-Champaign
"Accelerating Floating Flow Simulations using Physics-aware Data-driven Approaches"



Nathan Hesse
Data Scientist, Kaggle
"Open-source Crowdsourcing on Kaggle"



Alek Lerman
Stanford AI Lab
"Foundation Models for the Sciences"



Jun Sun Hong
Assistant Professor, University of Notre Dame
"Differentiable Hybrid Neural Modeling for Spatiotemporal Physics"



Pratik Pal
Sr. Research Scientist, Argonne National Labs



Russell Pollock
Professor, Stanford University



Shahank Hekmati
Senior Scientist, National Renewable Energy Laboratory
"Data-driven Modeling for Model Development: Data Sampling, Non-linear Models, and Uncertainty Quantification"



Yoosi Matias
Vice President, Google



Rossella Arzuffi
Associate Professor, Imperial College London



Yifang Ban
Professor, KTH Royal Institute of Technology



Ben Brackbill
Research Manager, University of Chicago



Marshall Burke
Associate Professor, Stanford University



Amir Sarraf-Farmani
Associate Professor, Carnegie Mellon University



Ian Foster
Professor, University of Chicago



Matthias Beme
Professor, Stanford University



Lu Lu
Assistant Professor, Yale University



Owen Price
Research Software Engineer, Google Labs



Russell Pollock
Professor, Stanford University



Riccardo Vignola
Associate Professor, KTH Royal Institute of Technology



Nathan Hesse
Data Scientist, Kaggle



Nils Thuery
Professor, Technical University of Munich



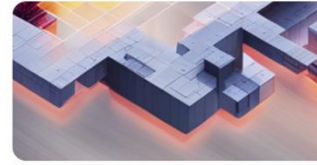
Riccardo Vignola
Associate Professor, KTH Royal Institute of Technology



Logan Ward
Computational Scientist, Argonne National Labs

2023 FLAME AI Challenge

Flow Physics Crowd-science Challenge for Turbulent Super-resolution



[Overview](#) [Data](#) [Code](#) [Models](#) [Discussion](#) [Leaderboard](#) [Rules](#)

Overview

Start

Sep 5, 2023

Close

Sep 14, 2023



Competition Host

Wai Tong Chung



Prizes & Awards

Kudos

Does not award Points or Medals

Participation

88 Entrants

51 Participants

39 Teams

814 Submissions

Description

Challenge Overview

Welcome to the FLAME AI 2023 Challenge!

Problem Statement

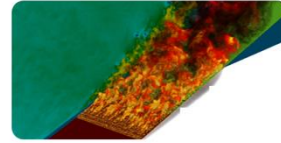
Participants in this Kaggle challenge will be provided with a dataset containing low-resolution 2D flowfield images. Your task is to develop state-of-the-art super-resolution models that can reconstruct high-resolution versions of these flowfields. The challenge lies in preserving the fine details and structures in the flowfields while increasing the resolution, which is critical for maintaining the accuracy of simulations and analyses.



#	Team	Members	Score	Entries	Last	Solution
1	ThomasX & Louis		0.00553	87	3y	
2	Anas Jnini		0.00620	28	3y	
3	nithin sekhar		0.00767	35	3y	
4	Thivin & Divij		0.00839	71	3y	
5	Varun Hiremath		0.00860	40	3y	
6	Abu Bucker		0.00891	20	3y	
7	Jordan & David		0.00898	39	3y	

2024 FLAME AI Challenge

Flow Physics Crowd-science Challenge for Spatio-temporal problems

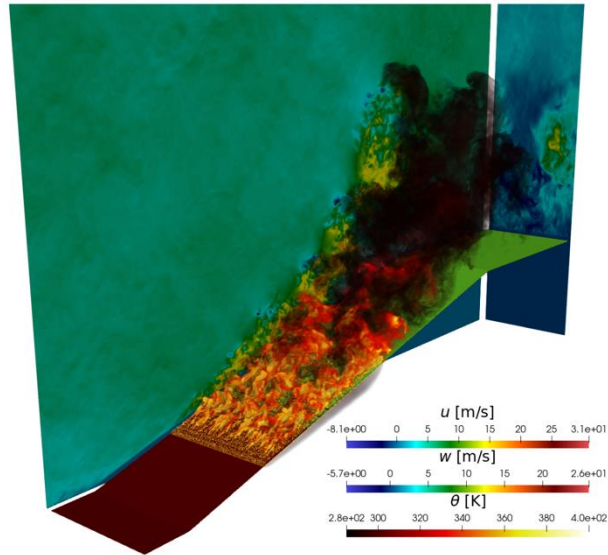


Overview

Data Code Models Discussion Leaderboard Rules

Overview

Welcome to the FLAME AI 2024 Challenge! The propagation of wildfires is a complex, dynamic process that is influenced by various factors, such as fuel, wind, terrain, and other environmental conditions. Accurately and reliably predicting the rate-of-spread of wildfires is of critical importance for fire management, rapid fire response, and fire mitigation.



Competition Host

BLASTNet



Prizes & Awards

\$5,000

Does not award Points or Medals

Participation

589 Entrants

190 Participants

172 Teams

2,533 Submissions

#	Team	Members	Score	Entries	Last	Solution
1	Ajay Asaithambi		0.00955	90	2y	
2	Jobayer Hossain		0.00960	28	2y	
3	Rafał Pawłowski		0.00978	71	2y	
4	zhuoqun		0.00987	40	2y	
5	jack robin		0.00998	13	2y	
6	nirmala 2343		0.01020	14	2y	
7	Moorthi E		0.01025	9	2y	

Summary

- Importance of diverse data for AI/ML tasks
 - BLASTNet: large, diverse, and high-fidelity dataset for turbulent flows
 - Can be used to develop and benchmark ML methods for solving problems – within and beyond flow physics
 - Benchmark results can aid the design of 3D super-resolution models in scientific and computer vision domains
- Community effort to contribute data for combustion AI/ML

Module 3: ML Methods

Machine Learning Types

Supervised learning	
<p>Relate labeled input to unknown output</p>	
Classification	Regression
<i>Logistic regression</i> <i>Classification trees</i> <i>Random forests</i> <i>Neural networks</i> <i>Support vector machines</i>	<i>Linear regression</i> <i>Regression trees</i> <i>Random forests</i> <i>Neural networks</i> <i>Gaussian processes</i>
Applications	
<ul style="list-style-type: none"> - Representation of fuel properties, thermochemical response functions, and potential energy surfaces - Parameterization of combustion manifolds - Prediction of risk occurrence and critical events - Combustion-closure modeling 	

Unsupervised learning	
<p>Identify pattern and discover structures</p>	
Clustering	Dimensional reduction
<i>Gaussian mixture models</i> <i>K-means</i> <i>Mean shift</i> <i>Spectral clustering</i>	<i>Principal component analysis</i> <i>Factor analysis</i> <i>Autoencoder</i> <i>Stochastic neighbor embedding</i>
Applications	
<ul style="list-style-type: none"> - Characterization of combustion regimes - Identification of low-dimensional manifolds - Discovery of structures and coherent features - Detection of anomalies and faults - Signal processing 	

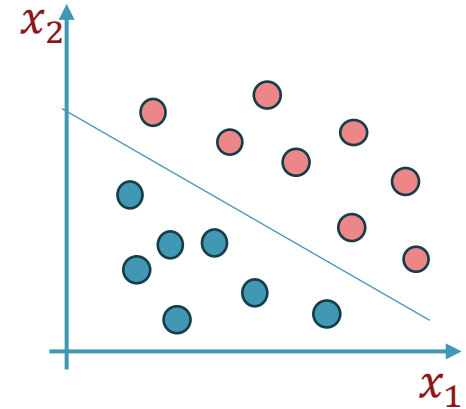
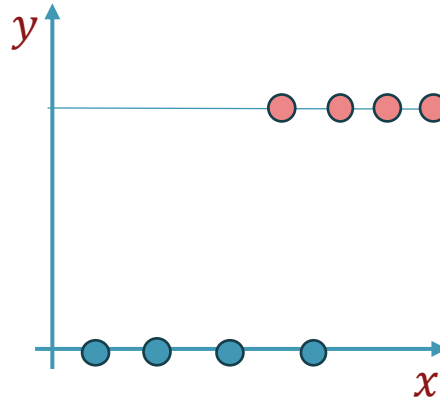
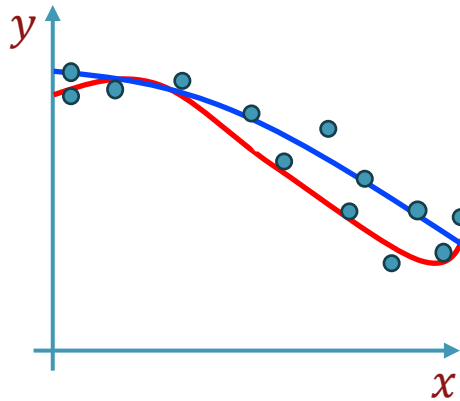
Semi-supervised learning	
<p>Learn from partially labeled data or through interaction with environment</p>	
Reinforcement learning	Generative approaches
<i>Q-learning</i> <i>State-action-reward-state-action</i> <i>Deep Q-learning</i> <i>Deep deterministic policy gradient</i>	<i>Generative adversarial network</i> <i>Variational autoencoders</i> <i>Boltzmann machine</i>
Applications	
<ul style="list-style-type: none"> - Optimization and control of combustion systems - Data augmentation and data generation - Generative combustion modeling - Robust combustion modeling - Operation with incomplete data 	

Machine Learning Types

- Supervised learning
 - Given: training data + desired outputs (labels)
- Unsupervised learning
 - Given: training data (without desired outputs)
- Semi-supervised learning
 - Given: training data + a few desired outputs
- Reinforcement learning
 - Rewards from sequence of actions

Supervised learning

- Learning from collection of labeled input/output data, $\{(x, y)_i\}_{i=1}^N$, to predict outcome: $\hat{y} = f(\hat{x})$
- Learning tasks
 - Regression: Predicting continuous function (y : continuous values)
 - Classifications: Predicting discrete labels (y : categorical)



Supervised learning

- Algorithms and methods
 - Logistic regression
 - Decision tree and random forest
 - Neural networks
 - Support vector machines

Supervised learning methods

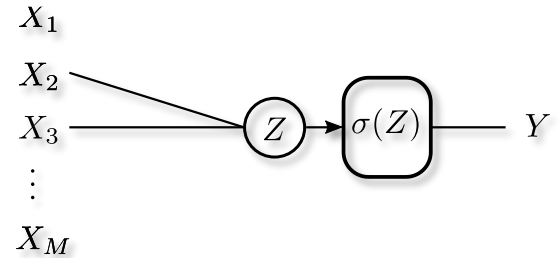
- Logistic regression
 - Binary classification of learning data $\{(x, y)_i\}_{i=1}^N$ with $y_i \in \{\psi_1 = 0, \psi_2 = 1\}$
$$Y(X) = P(Y = \psi_1 | X) = \sigma(Z)$$

$$Z = w^T X + b$$

with non-linear transfer function

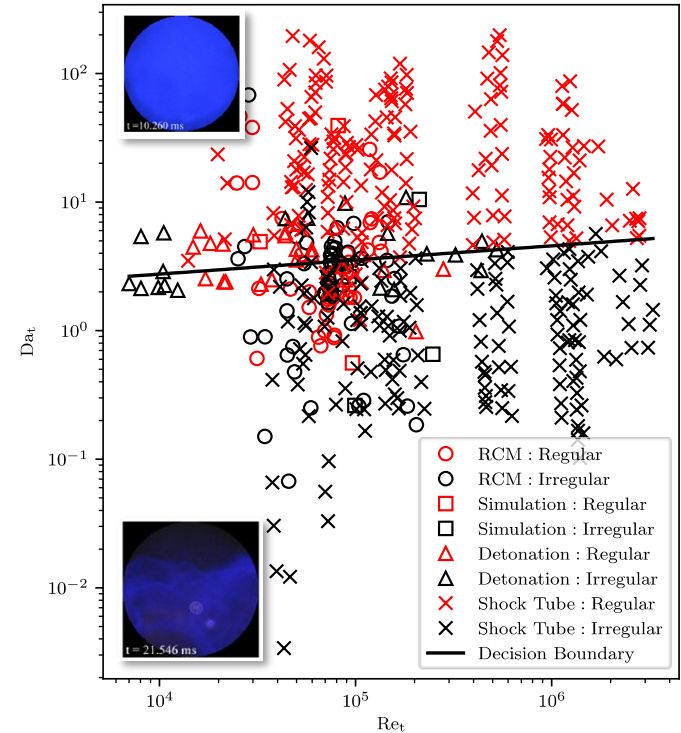
$$\sigma(Z) = \frac{1}{1 + \exp\{-Z\}}$$

- Determine model parameters $\theta = (w, b)^T$ by minimizing likelihood function



Supervised learning methods

- Logistic regression
 - Application
 - Classification of irregular combustion ignition conditions

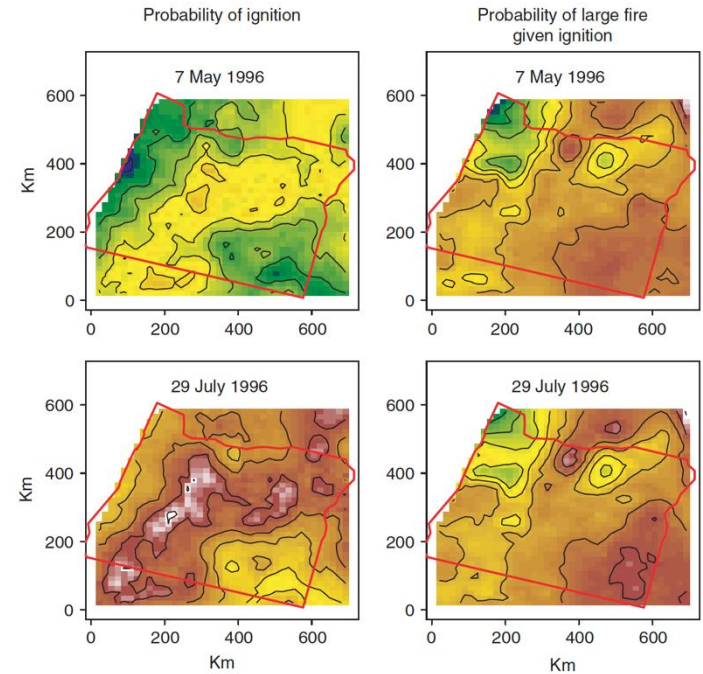


Supervised learning methods

- Logistic regression

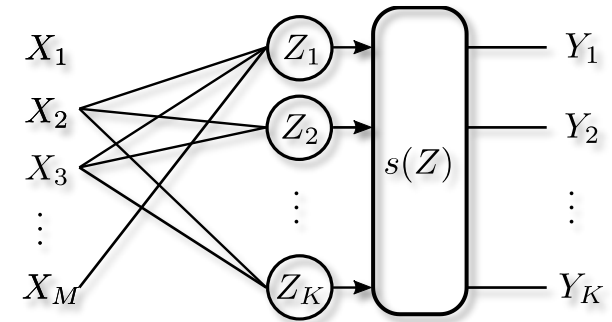
- Application

- Classification of irregular combustion ignition conditions
 - Prediction of fire occurrence and fire risk assessment



Supervised learning methods

- Logistic regression
 - Remarks and limitations
 - Pre-determined hypothesis
 - Fundamental building block of neural networks
 - Linear separability of feature space
 - Limitation in representing non-linear decision boundaries



Supervised learning methods

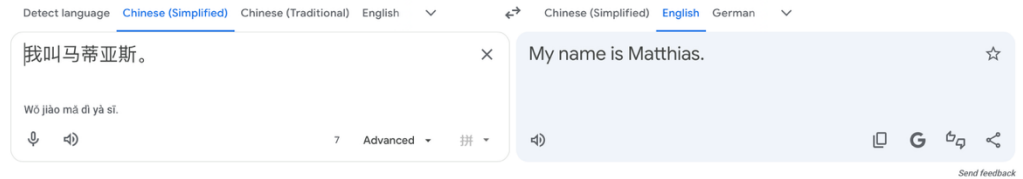
- Classification and Regression Trees (CARTs)
 - Popular Data types

Images/Spatial (N-D)



CNNs, Vision Transformers

Language/Sequential (1D)



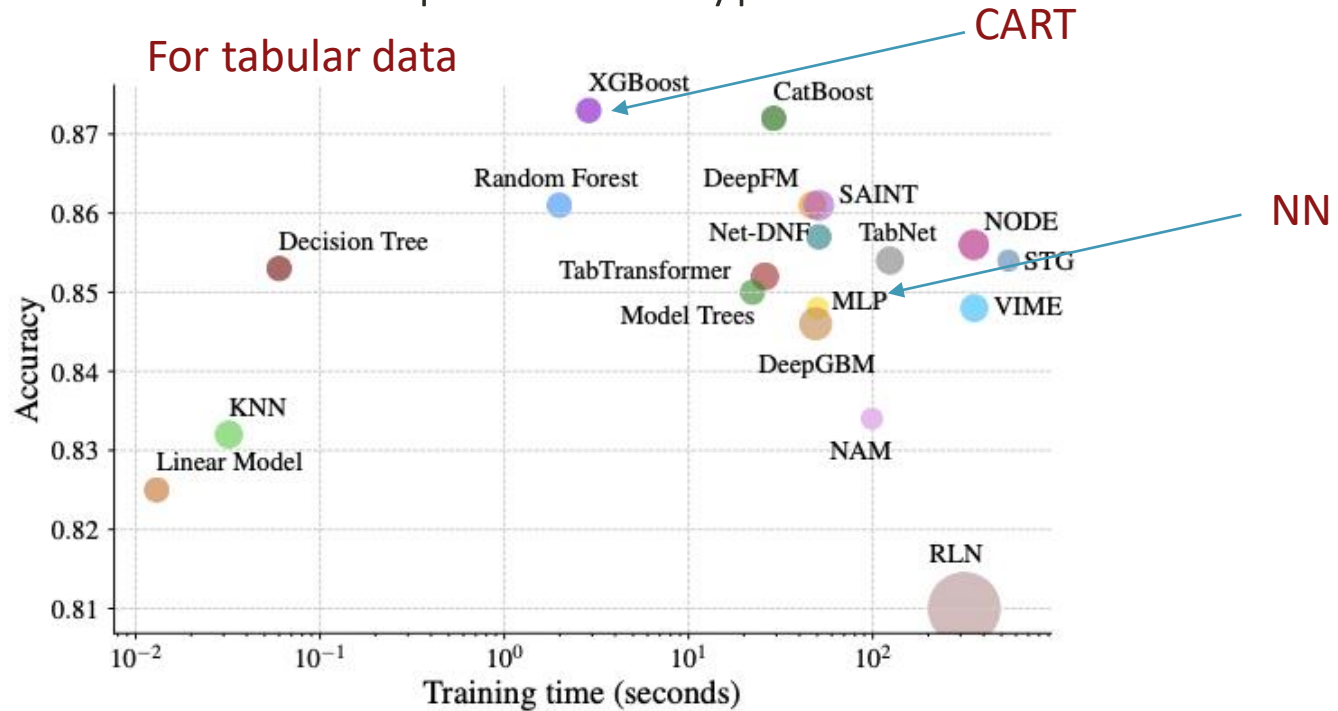
RNNs, Transformers, LSTMs

Tabular Data (0D)



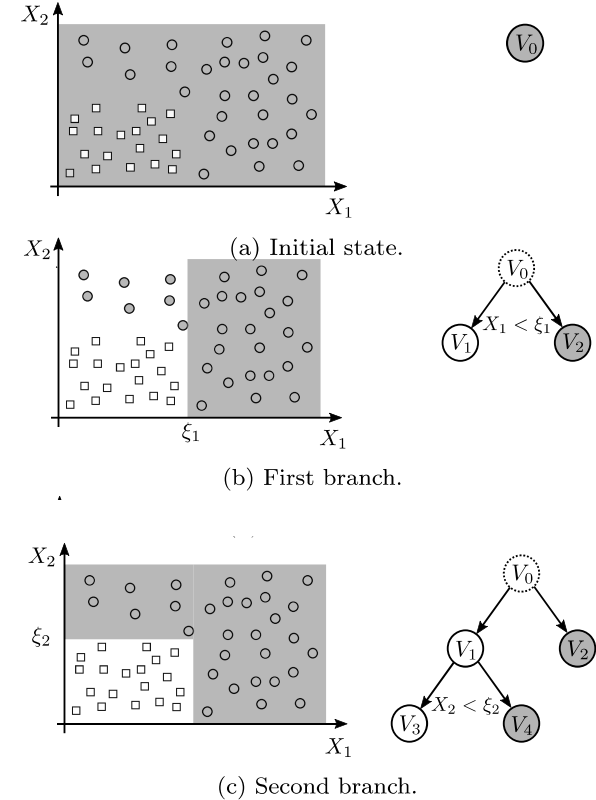
Supervised learning methods

- Why care about CARTs? Popular Data types



Supervised learning methods

- Training a decision tree
 - Regression in logistic regression/deep learning
 - Find slope of function (weights) that minimize **error** between predicted and target variable.
 - Classification in logistic regression/deep learning
 - Find slope of function (weights) that maximizes the likelihood of predicted variable.
 - Regression in a decision tree
 - Find a split that maximizes “purity”
 - Related to variance
 - Classification in a decision tree
 - Find a split that maximizes “information gain”
 - Related to entropy

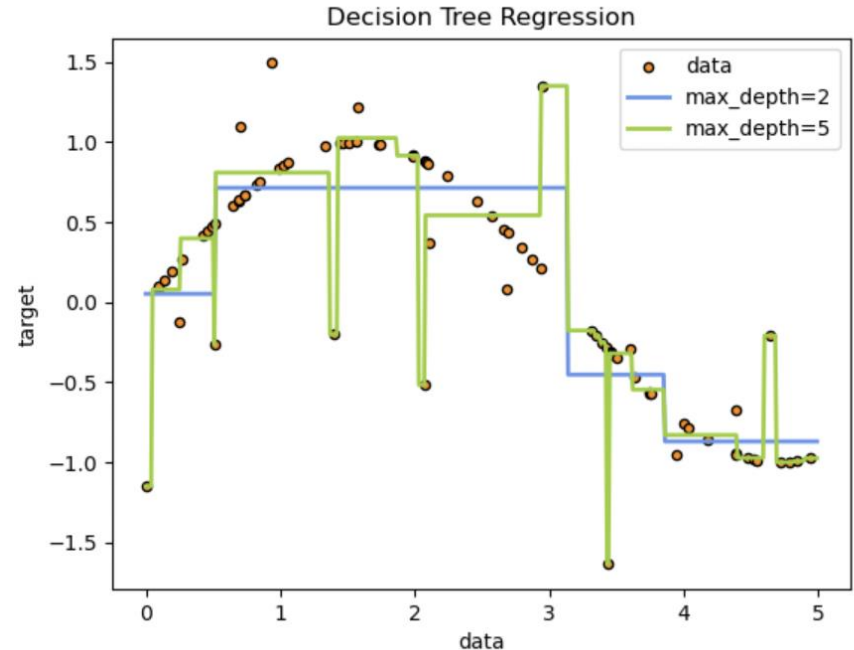
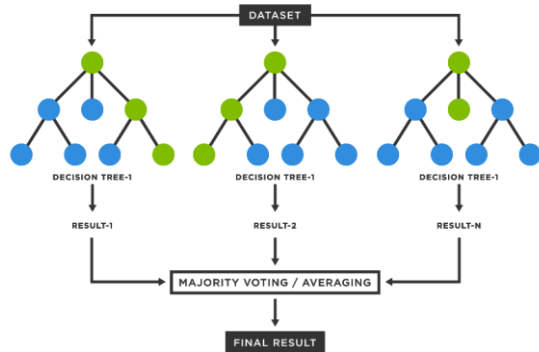


Can inspect node structures to interpret model behavior -> Feature importance score

Supervised learning methods

• Decision Tree Regression

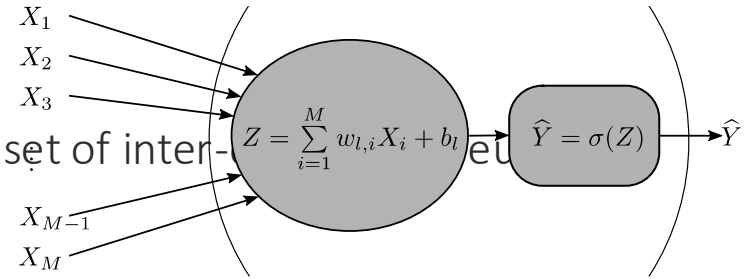
- Decision tree regressors are may be less suitable for smooth functions
- Not suitable for extrapolation
- Prone to overfitting
- Rectify with ensemble methods



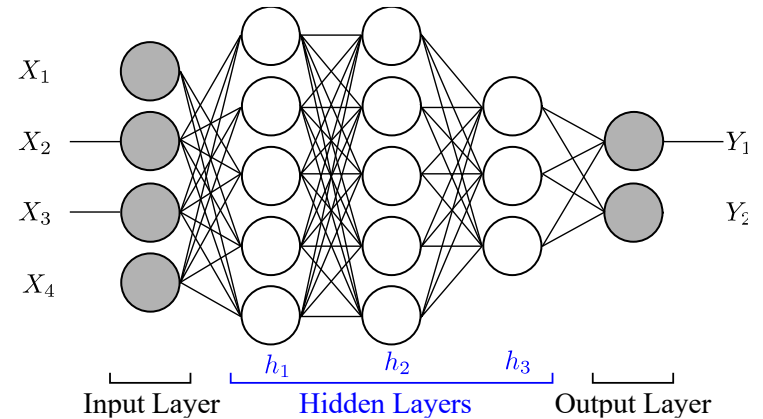
Supervised learning methods

- Neural Networks

- Mimics neural brain structure through dense set of inter-
- Fundamental building block: logistic regression → Multilayer perceptrons



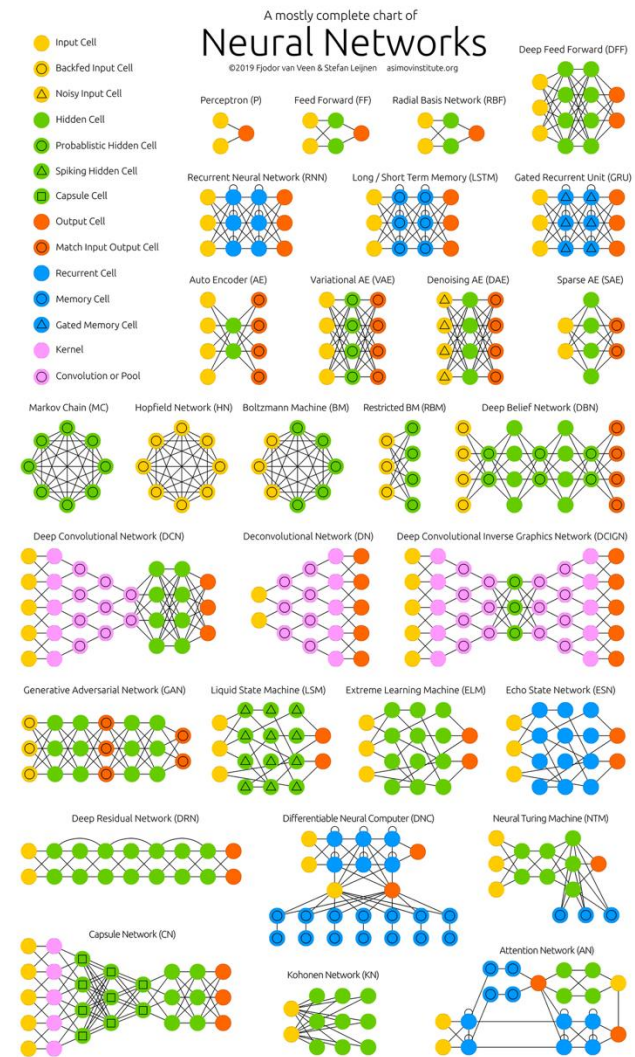
$$\sigma(Z) = \frac{1}{1 + \exp\{-Z\}}$$



Supervised learning methods

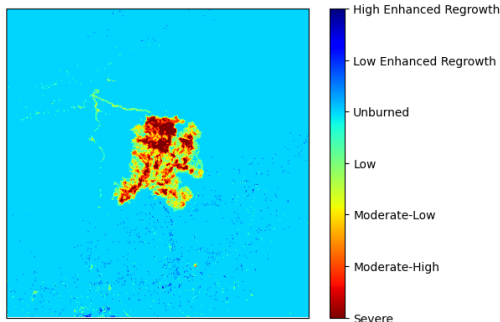
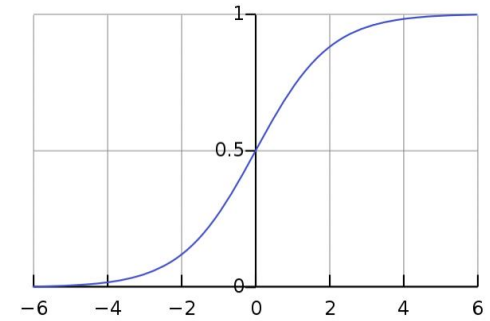
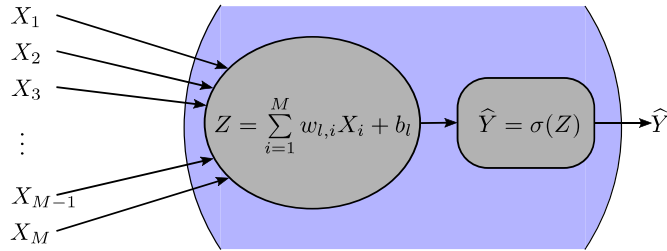
- Neural Networks

- Specialized network architectures
 - Deep neural networks
 - Convolutional neural networks
 - Recurrent neural networks
 - Graph neural networks



Supervised learning methods

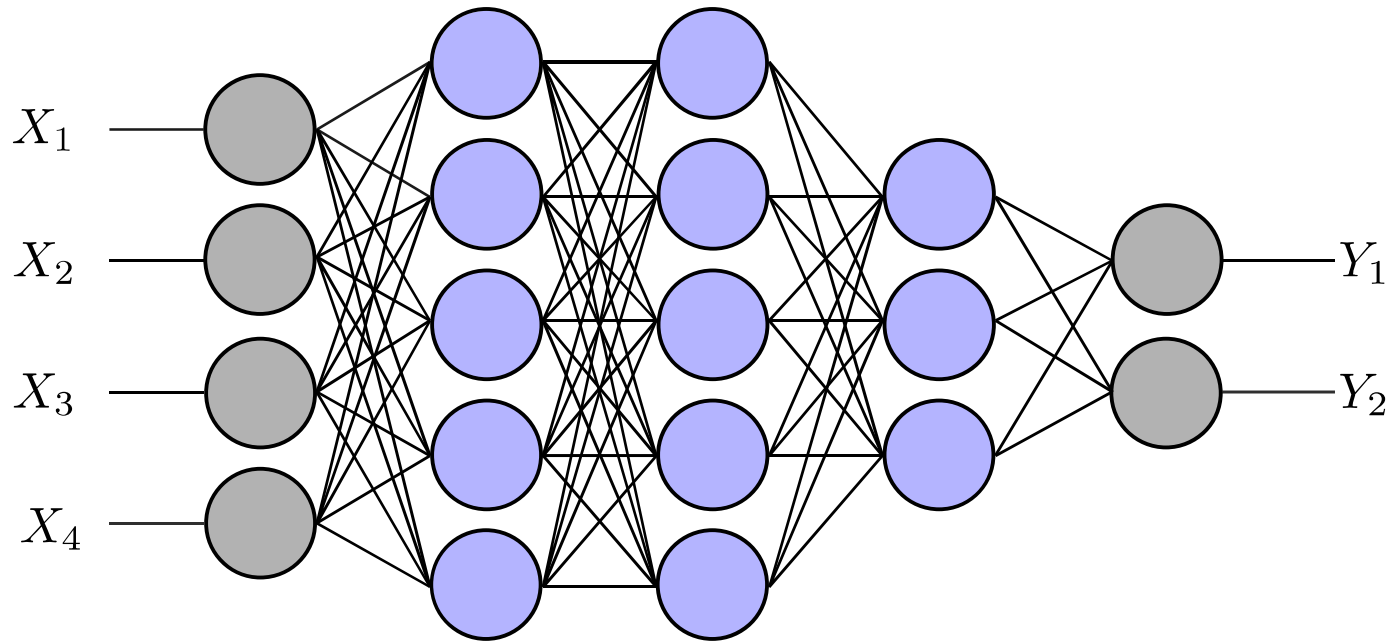
- So far we've only discussed binary classification.
- How to do multiclass classification?



Class	Labels						
High Regrowth	1	0	0	0	0	0	0
Low Regrowth	0	1	0	0	0	0	0
Unburned	0	0	1	0	0	0	0
Low	0	0	0	1	0	0	0
Moderate-Low	0	0	0	0	1	0	0
Moderate-High	0	0	0	0	0	1	0
Severe	0	0	0	0	0	0	1

Supervised learning methods

- Neural Networks – A bunch of connected logistic regression nodes



Supervised learning methods

- Deep learning

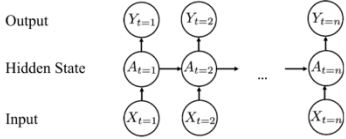
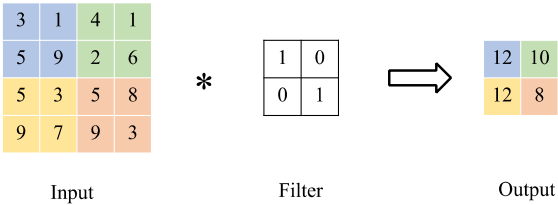
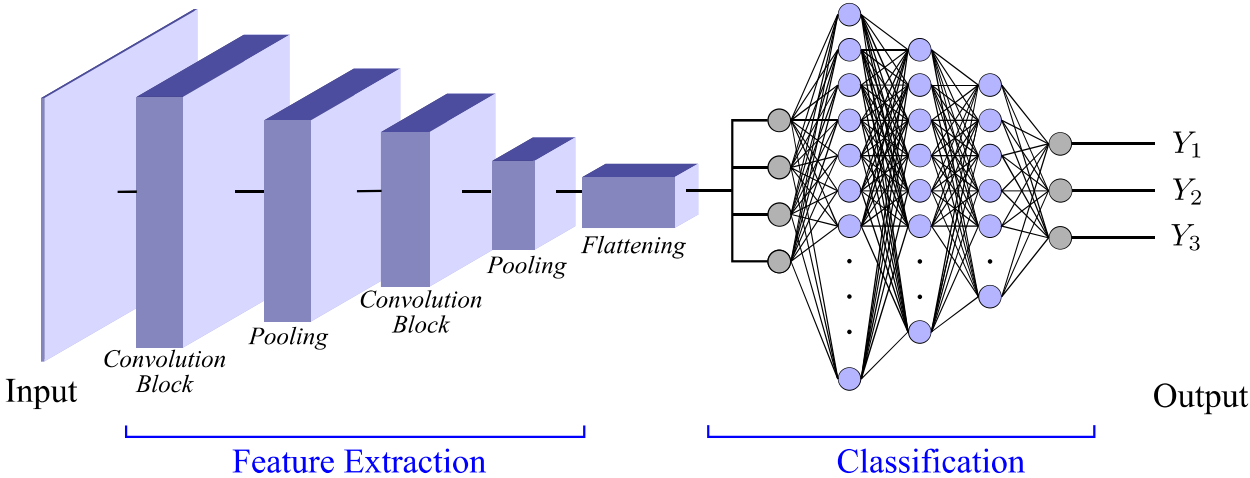
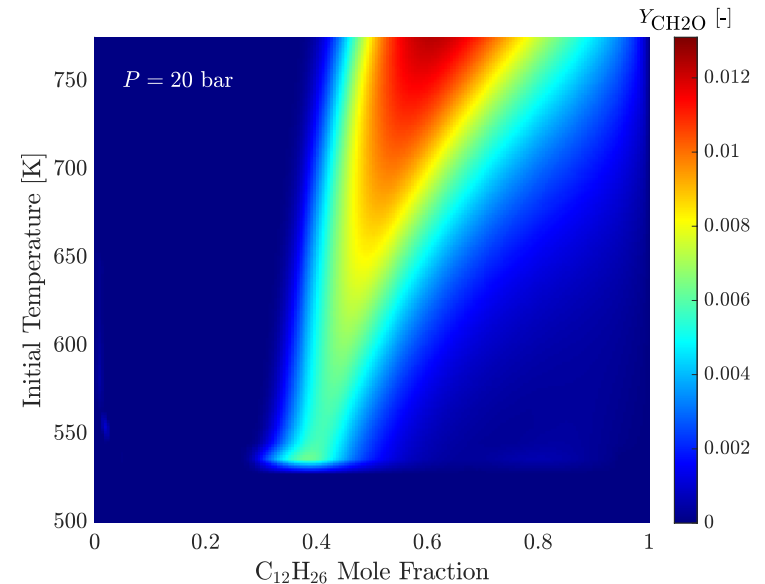
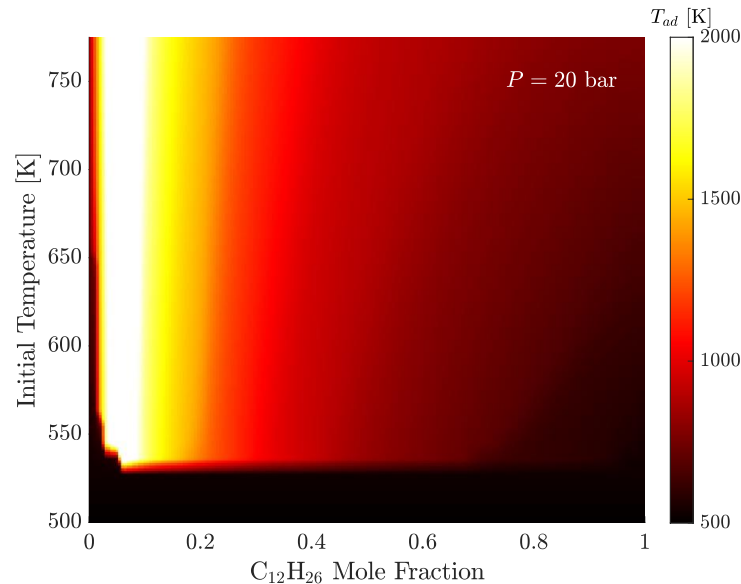


Fig. 16. Schematic of a many-to-many RNN.

Supervised learning methods

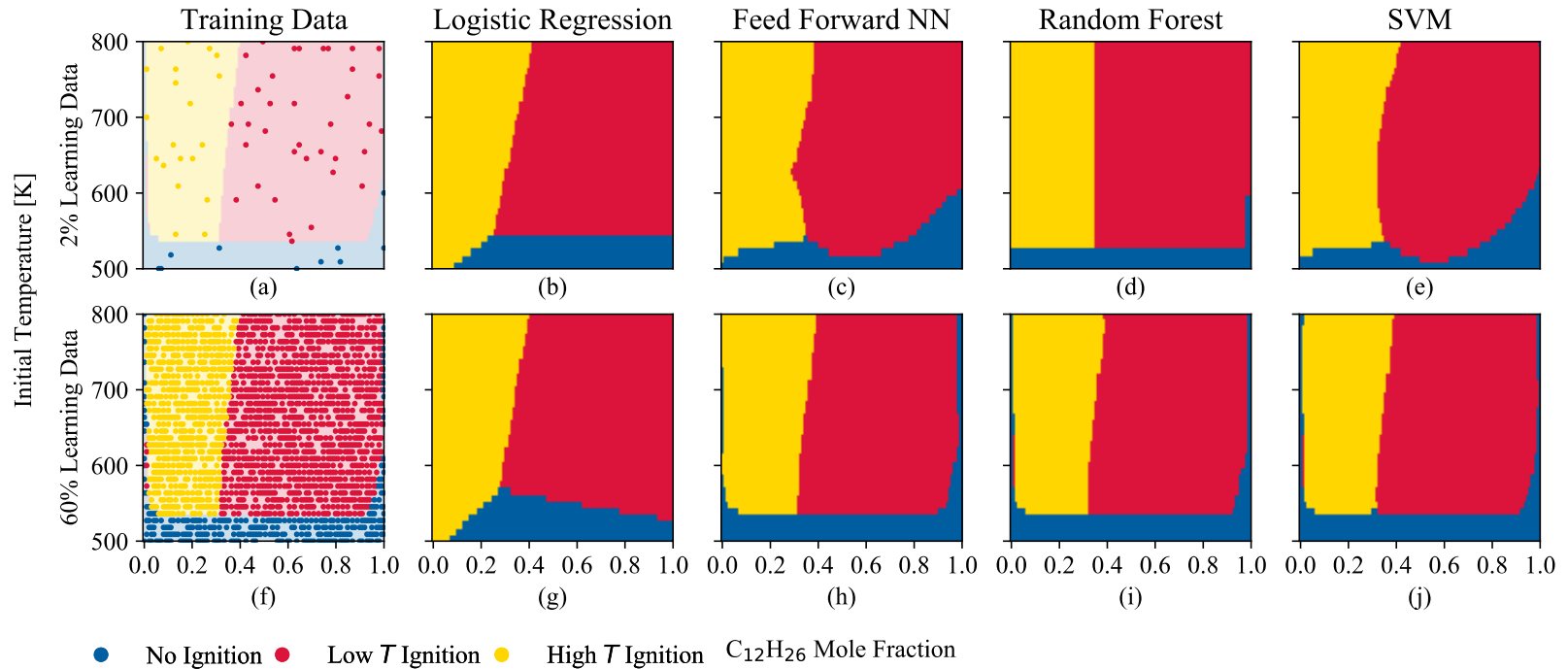
- Classification of ignition behavior in $C_{12}H_{26}$ /air mixture



https://github.com/lhmeGroup/CombML_Tutorials

Supervised learning methods – Data!

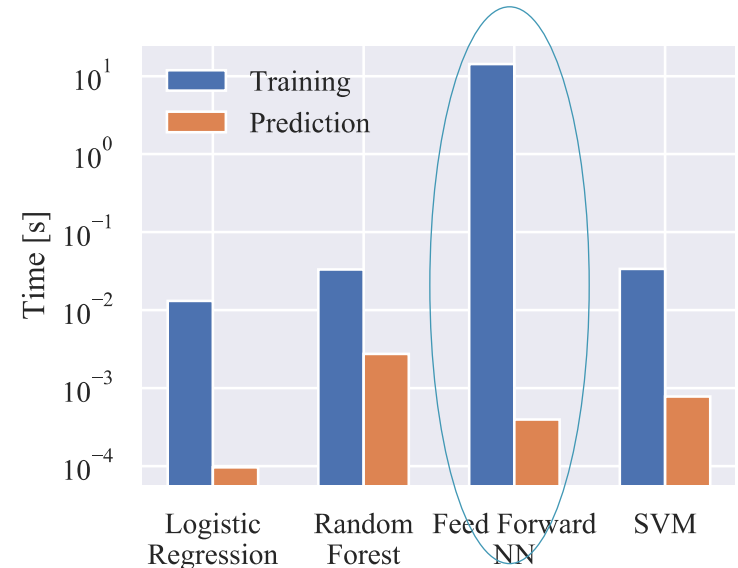
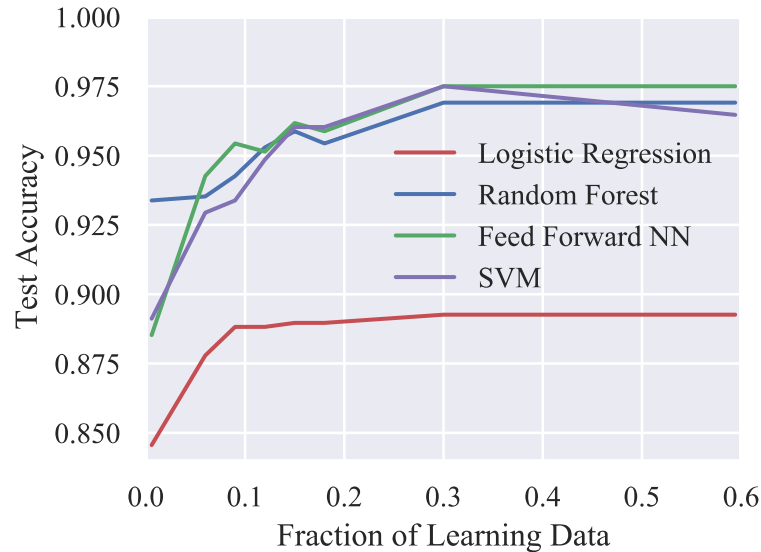
- Classification of ignition behavior in $C_{12}H_{26}$ /air mixture



https://github.com/lhmeGroup/CombML_Tutorials

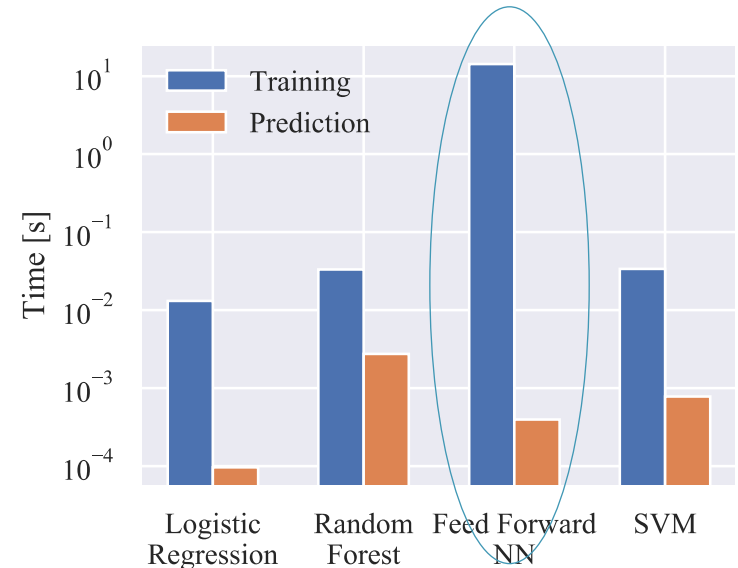
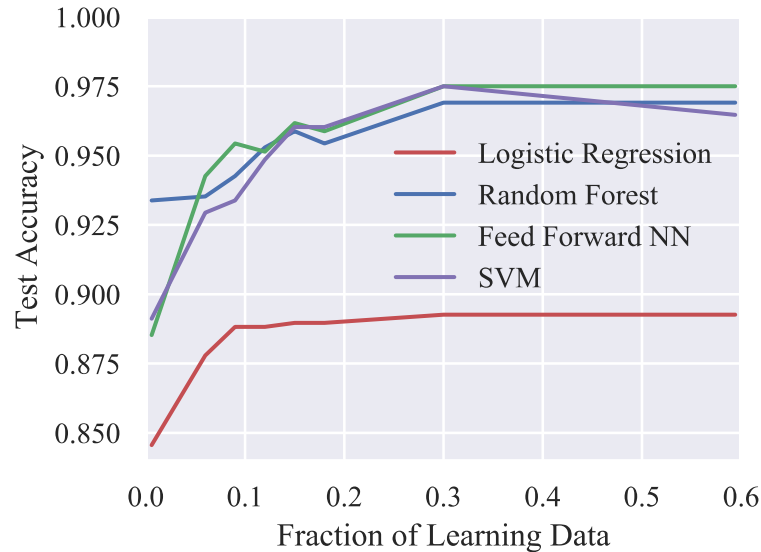
Supervised learning methods – Cost!

- Classification of ignition behavior in $C_{12}H_{26}$ /air mixture



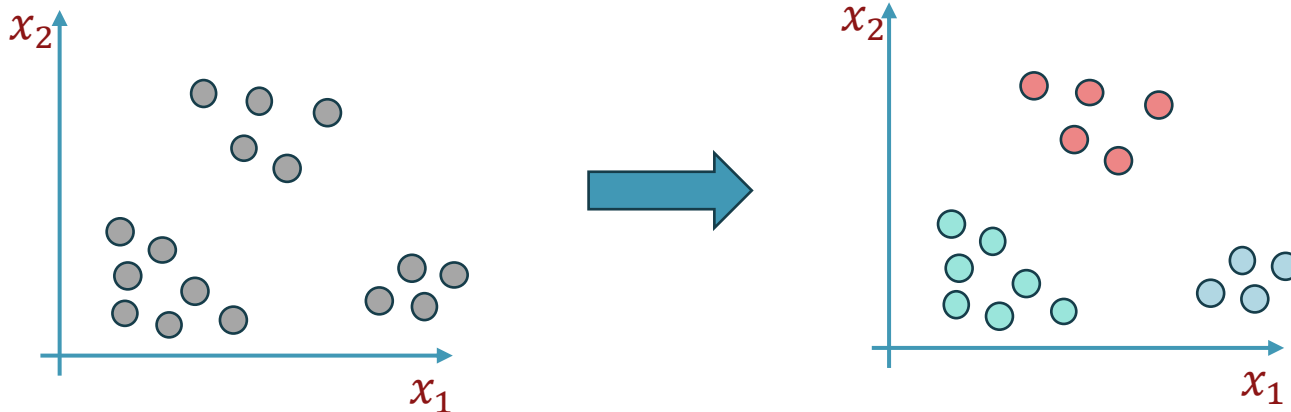
Supervised learning methods – Cost!

- Classification of ignition behavior in $C_{12}H_{26}$ /air mixture



Unsupervised learning

- Learning from collection of input data, $\{x_i\}_{i=1}^N$ (without labels)
- Learn underlying structure
 - Clusters hidden in the dataset
 - Outliers: particularly unusual and/or interesting datapoints.
 - Useful signal hidden in noise



Unsupervised learning

- Algorithms and methods
 - Clustering
 - K-mean clustering
 - Hierarchical clustering
 - Spectral clustering
 - Dimensional reduction
 - Principal component analysis
 - Non-negative matrix factorization
 - Locally linear embedding

Unsupervised learning methods

- K-mean clustering

- Subdivision of dataset with N points into K clusters by minimizing cost function

$$J = \sum_{i=1}^N \sum_{k=1}^K w_{ik} \|x_i - \mu_k\|^2$$

- Strengths:

- Simple
- Suitable for small K
- Efficient ($O(KNn)$, n : iteration) \rightarrow linear algorithm

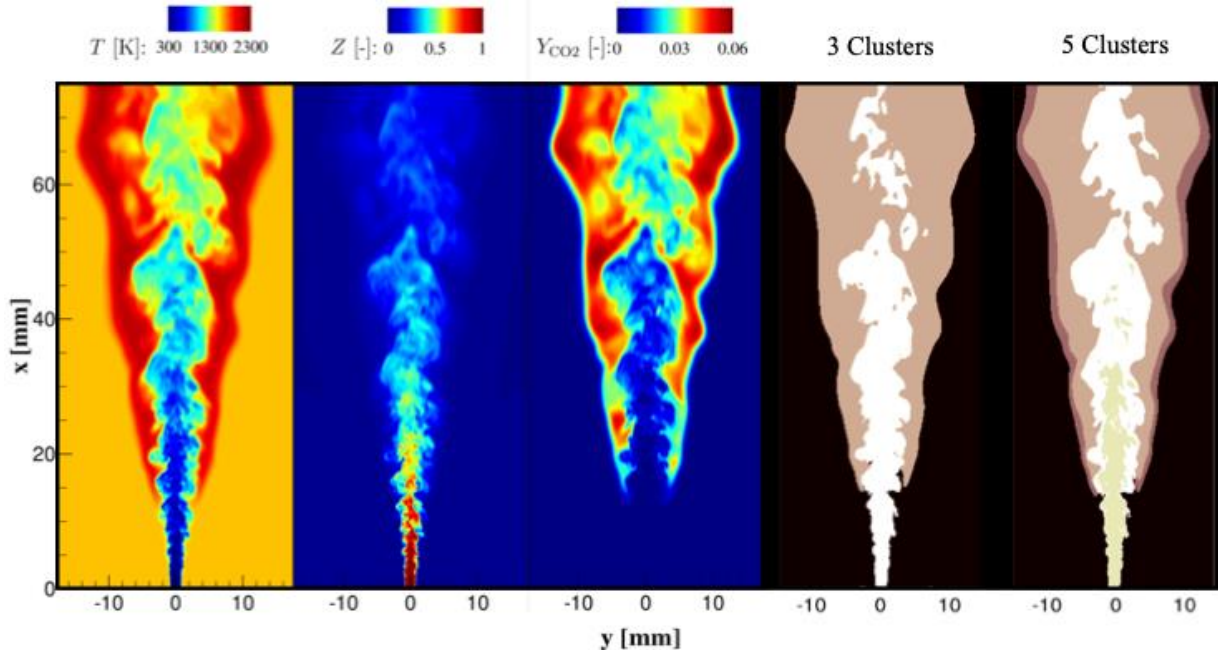
- Weakness:

- local optimum, dependent in initialization
- Sensitive to outliers (fix: remove outliers; random sampling)

Unsupervised learning methods

- K-mean clustering

- Example: 7-dimensional feature space: $x = \mathcal{F}(T, Z, Y_{CO_2}, Y_{H_2O}, Y_{CO}, Y_{CH_4}, Y_{O_2})$



3 clusters:

- Reactants
- Products
- Oxidizer

5 cluster:

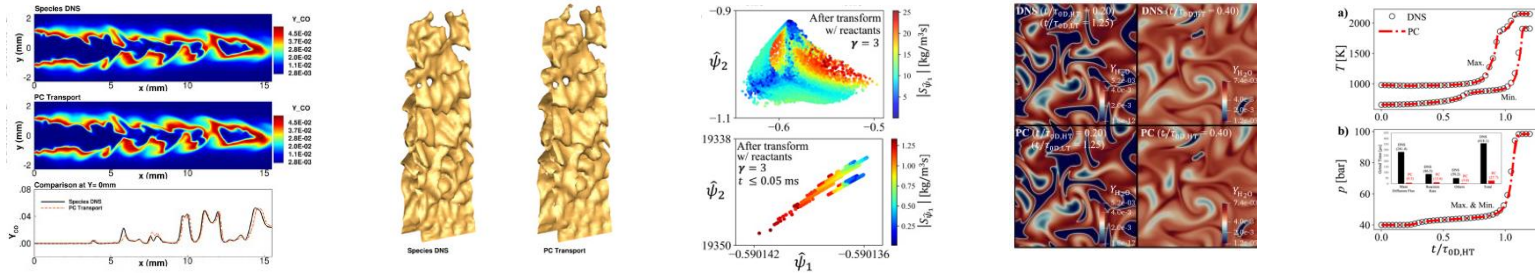
- Distinguish reaction zone
- Subdivide reactant region in low-temp. fuel region and intermediate-temp. region

Unsupervised learning methods

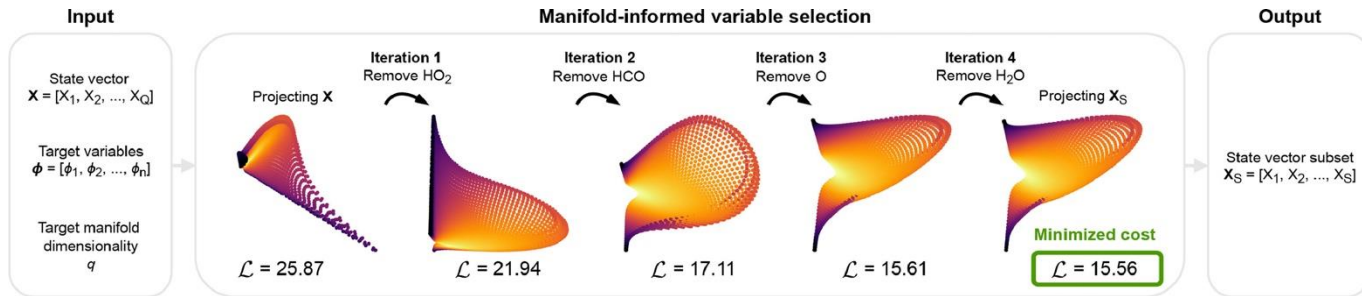
- Principal Component Analysis (PCA)
 - Mathematical approach to reveal preferential directions in multidimensional datasets through identifications of correlation in state space
 - Decomposition of covariance matrix $S = \frac{1}{N-1} x^T x = Q\Lambda Q^T$
 - Principal component score $P = xQ \rightarrow x \simeq x_m = P_m Q_m^T$ (m : retained eigenmodes)
 - Benefits
 - Linearity of principal components
 - Control of accuracy through number of retained modes m
 - Applications
 - Low-dimensional combustion modeling
 - Parameterization and identification of combustion manifolds
 - Flame classification

Unsupervised learning methods

- Principal Component Analysis (PCA)
 - Dimensional reduction to accelerate combustion simulations

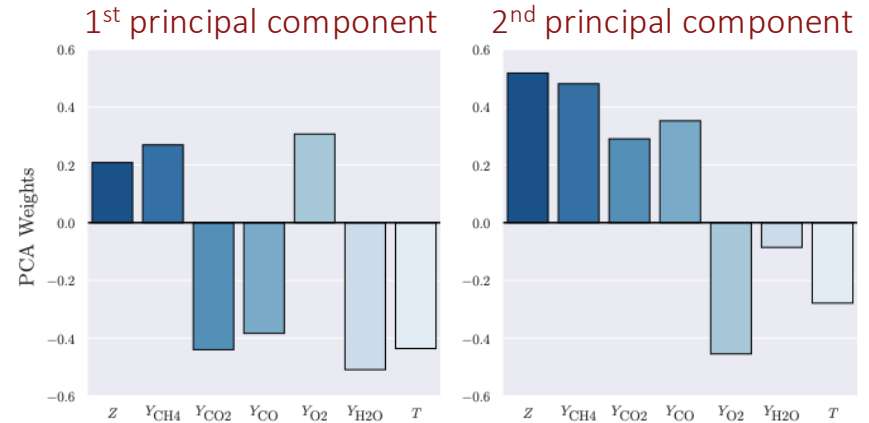
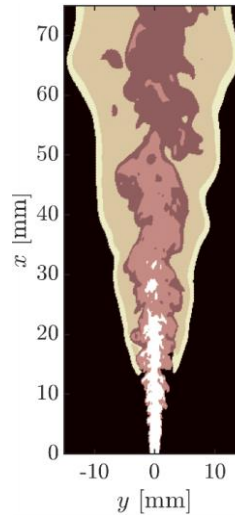
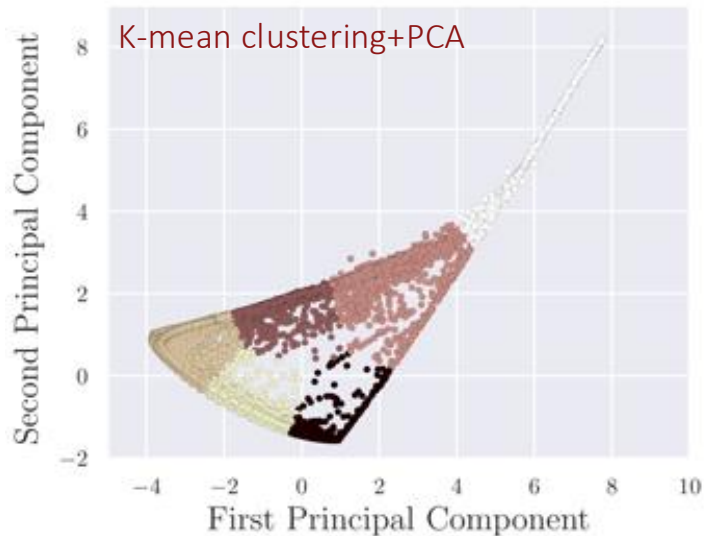


- Dimensional reduction, automatic identification of combustion manifolds



Unsupervised learning methods

- Principal Component Analysis (PCA)



Semi-supervised learning and generative models

- Combine labeled $((X_L, Y_L) = \{(x, y)_i\}_{i=1}^L)$ and unlabeled $X_u = \{x_i\}_{i=L+1}^N$ data with $(L \ll N)$

Semi-supervised learning and generative models

- Estimate probabilistic process that generates a set of observations \mathcal{D}
 - $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$: supervised generative models learn joint distribution $p(x_i, y_i)$, often to compute $p(y_i|x_i)$
 - $\mathcal{D} = \{x_i\}_{i=1}^N$: unsupervised generative models learn distribution through

- Direct estimation of $p(x_i)$: chose parameterized family $p(x|\theta)$; learn θ by maximizing log-likelihood:

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^N \log p(x_i|\theta)$$

- Latent variable model: Introduce latent state y_i and estimate $p(x_i, y_i)$; introduce joint distribution $p(x, y|\theta)$; learn θ by maximizing log-marginal likelihood:

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^N \int \log p(x_i, z_i|\theta) dz$$

→ Require simple evaluation of $p(x_i|\theta)$

Semi-supervised learning and generative models

- Generative algorithms
 - Self-training
 - Gaussian Mixture Model (GMM)
 - Normalizing flows
 - Variational autoencoder
 - Generative Adversarial Network (GAN)
 - Diffusion models

Semi-supervised learning and generative models

- Self-training
 - Train f from (X_l, Y_l)
 - Predict on $x \in X_u$
 - Add $(x, f(x))$ to labeled data
 - Repeat
- Properties
 - Simplest semi-supervised learning method
 - Requires high confidence in accurate predictions
 - Amplification of early mistakes
 - Convergence properties unclear (except for special cases: expectation minimization, linear problems)

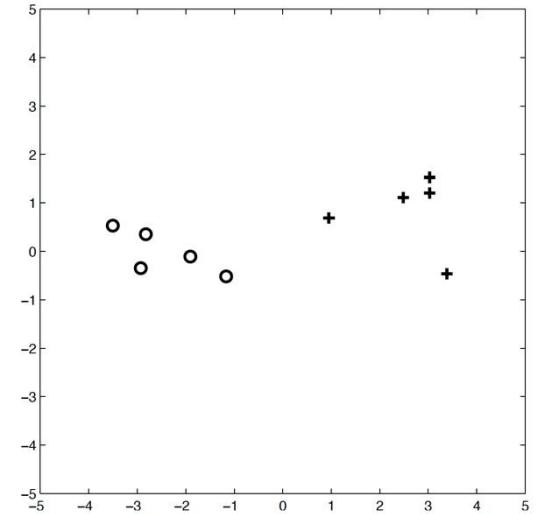
Semi-supervised learning and generative models

- Generative models: Gaussian Mixture Model (GMM)
 - Labeled data: (X_L, Y_L)
 - Model parameters $\theta = \{w_1, w_2, \mu_1, \mu_2, \sigma_1, \sigma_2\}$
 - Assume data follows presumed distribution (Gaussian)
 - GMM Bayesian inference:

$$\begin{aligned} p(x, y|\theta) &= p(y|\theta)p(x|y, \theta) \\ &= w_y \mathcal{N}(x; \mu_y, \sigma_y) \end{aligned}$$

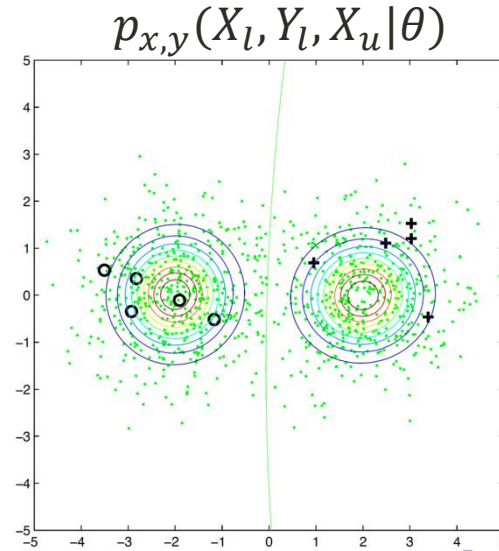
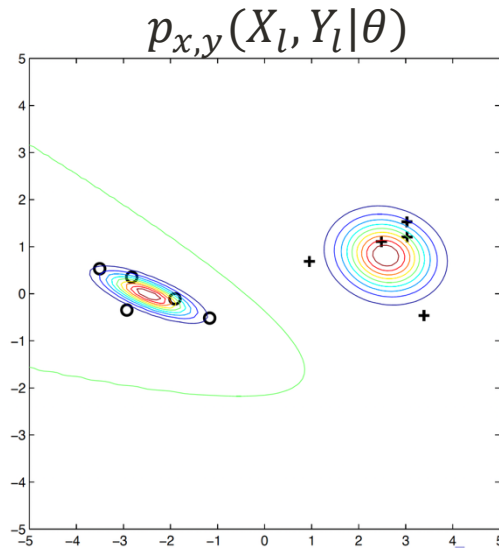
- Classification

$$p(y|x, \theta) = \frac{p(x, y|\theta)}{\int p(x, y'|\theta) dy'}$$



Semi-supervised learning and generative models

- Generative models: Gaussian Mixture Model (GMM)
 - Labeled data: (X_L, Y_L)
 - Model parameters $\theta = \{w_1, w_2, \mu_1, \mu_2, \sigma_1, \sigma_2\}$



Semi-supervised learning and generative models

- Normalizing flows

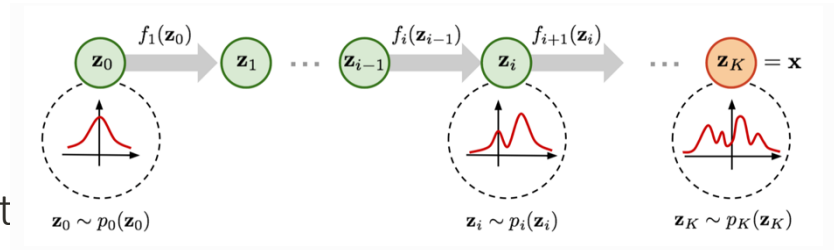
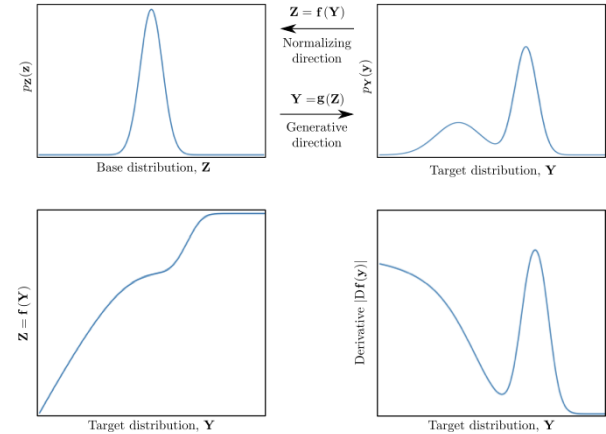
- Recall: $p_y(Y)dY = p_z(Z)dZ \rightarrow p_y(Y) = p_z(Z) \frac{dZ}{dY}$

- Sequence of non-linear transformations to a simple distribution ($Z = f(Y) \leftrightarrow Y = g(Z)$)

$$\begin{aligned}
 p_y(Y) &= p_z(f(Y)) \left| \det \left(\frac{df}{dY} \right) \right| \\
 &\text{pushforward} \\
 &= p_z(f(Y)) \left| \det \left(\frac{dg}{dZ} \right) \right|^{-1}
 \end{aligned}$$

- Combine multiple invertible function $f_{1,\dots,K}$ to

represent more complex distributions



Kobyzev et al. Normalizing Flows: An Introduction to the Normalizing Flow Architecture. *Pattern Anal. Mach. Intell.*, 2020

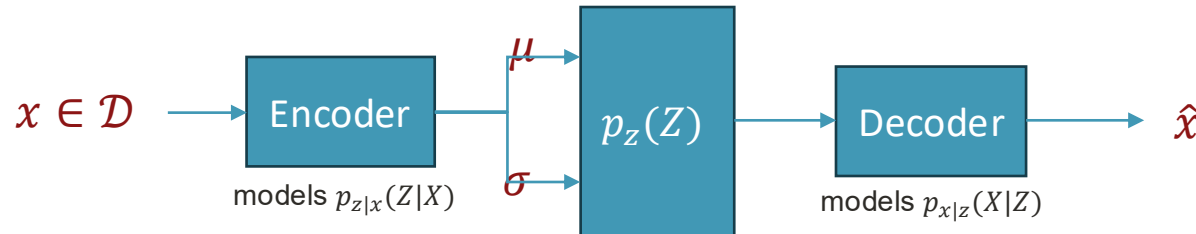
Lippe: Normalizing Flows for image modeling, https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial11/NF_image_modeling.html

Semi-supervised learning and generative models

- Generative Adversarial Networks (GANs)
 - class of density-free generative models with (nearly) unrestricted generator functions
 - Adversarial discriminator networks allows GANs to learn by minimizing the Jensen-Shannon divergence
 - Generate samples from unlabeled data $\mathcal{D} = \{x_i\}_{i=1}^N$

Semi-supervised learning and generative models

- Generative Adversarial Networks (GANs)
 - Ingredients
 - Discriminative models: learn conditional distribution $p_{y|x}(Y|X)$
 - Generative models: learns joint distribution $p_{x,y}(X, Y)$
 - Basic building block: Variational autoencoder (VAE): $p_x(X, Z|\Theta) = p_x(X|f_\theta(z))p_z(Z)$



Semi-supervised learning and generative models

- Generative Adversarial Networks (GANs)

- VAEs vs GANs

VAEs

- Minimize KL-divergence
- Minimize bound on divergence btw. generated distribution and target distribution
- Simpler optimization; faster and more reliable training
- Blurry results

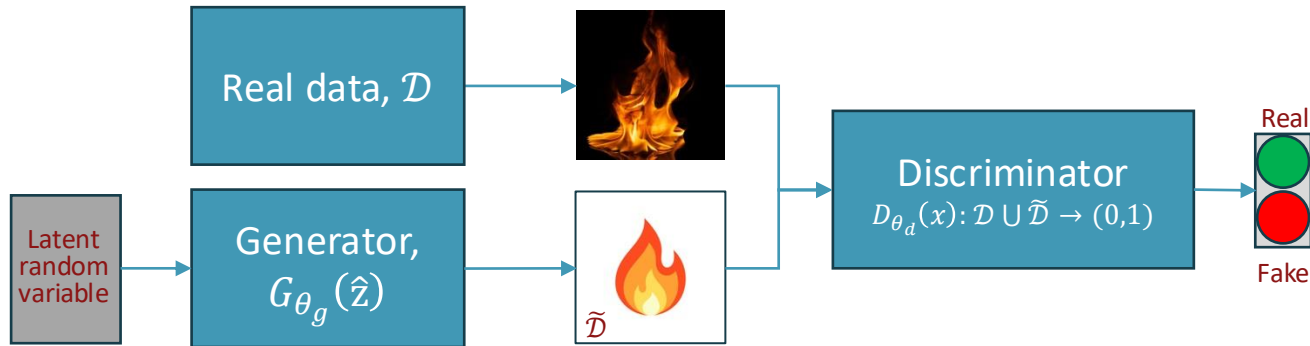
GANs

- Minimize Jensen-Shannon divergence
- Minimize divergence btw. generated distribution and target distribution
- Noisy and difficult to optimize
- Sharp results

Semi-supervised learning and generative models

- Generative Adversarial Networks (GANs)

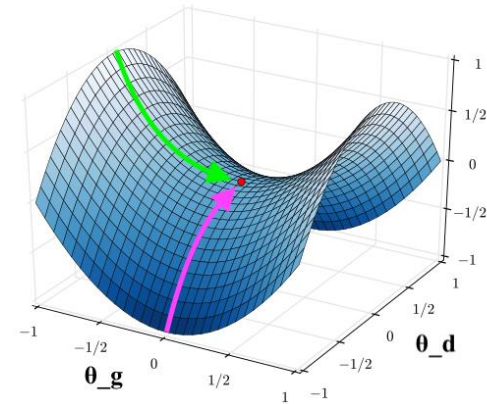
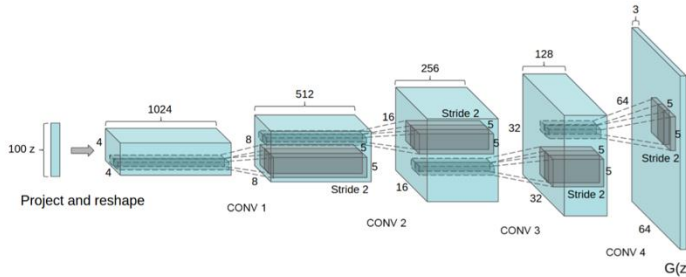
- Use unrestricted generator $G_{\theta}(z): p_x(X|\Theta_g) = p_z(Z)$ with $z = G_{\theta}^{-1}(x)$
- Learn by comparing model samples with examples from \mathcal{D}
 - Sampling from generator: $\hat{x} = G_{\theta_g}(\hat{z})$ where $\hat{z} \sim p_z(Z)$
 - Discriminator tries to distinguish \hat{x} from true sample data $\mathcal{D}(x) = \Pr(x \sim p_x(X))$



Semi-supervised learning and generative models

- Generative Adversarial Networks (GANs)
 - Saddle point optimization: learn $G_{\theta_g}(z)$ and $D_{\theta_d}(x)$ jointly

$$\min_{\theta_g} \max_{\theta_d} \underbrace{\mathbb{E}_{p_{data}} [\log D_{\theta_d}(x)]}_{\text{Likelihood of true data}} + \underbrace{\mathbb{E}_{p_z} \left[\log \left(1 - \log D_{\theta_d} \left(G_{\theta_g}(z) \right) \right) \right]}_{\text{Likelihood of generated data}}$$

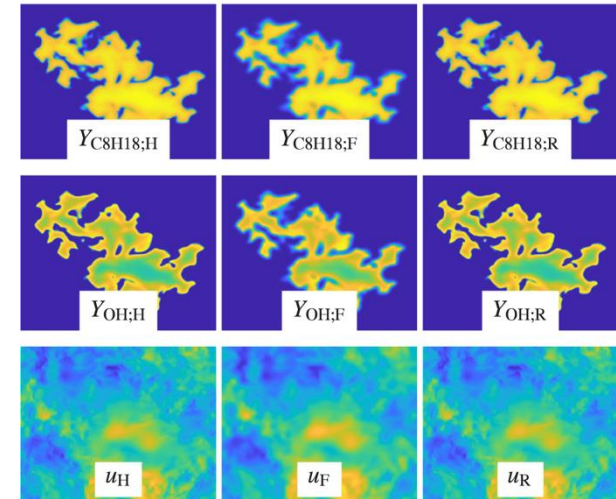
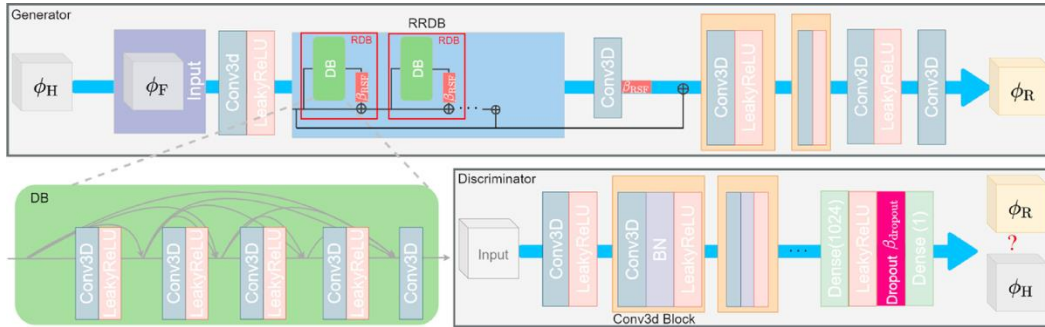


Semi-supervised learning and generative models

- Generative Adversarial Networks (GANs)
 - Problems with GANs
 - Vanishing gradients: discriminator becomes “too good” and generator gradient vanishes
 - Non-Convergence: generator and discriminator oscillate without reaching an equilibrium.
 - Mode Collapse: generator distribution collapses to a small set of examples
 - Mode Dropping: the generator distribution doesn't fully cover data distribution

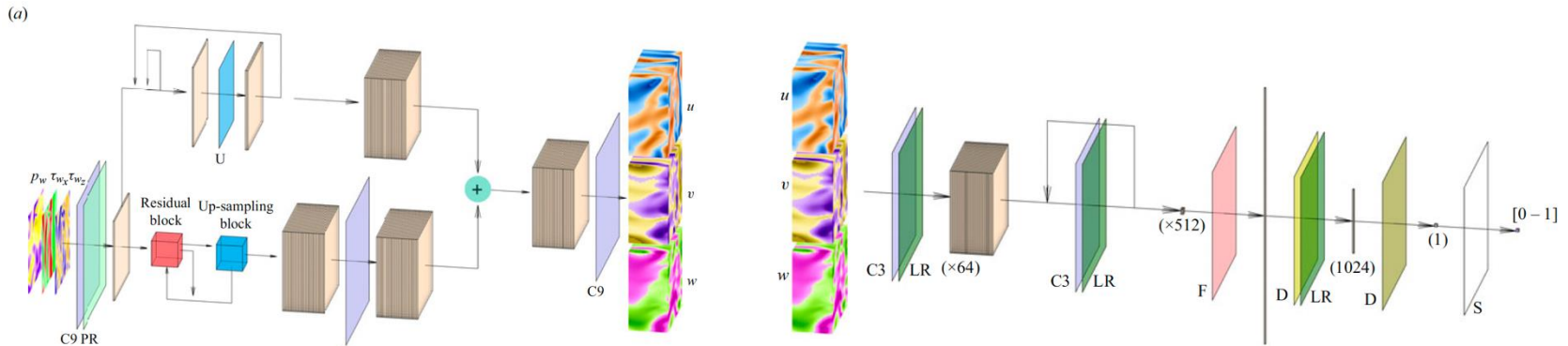
Semi-supervised learning and generative models

- Generative Adversarial Networks (GANs)
 - Application: superresolution modeling for subgrid-closure modeling [physics-informed enhanced super-resolution GAN (PIESRGAN)]



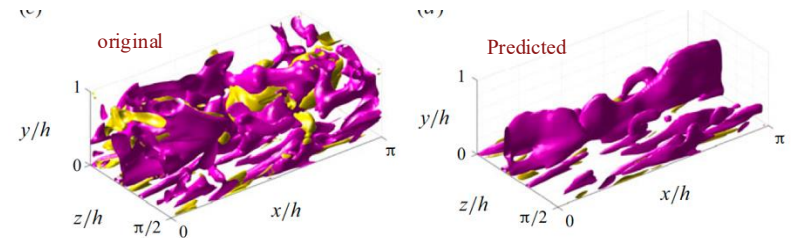
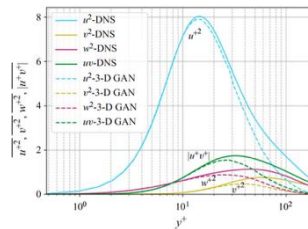
Semi-supervised learning and generative models

- Generative Adversarial Networks (GANs)
 - Application: 3D GANs for turbulent flow estimation from wall measurements



Generator network

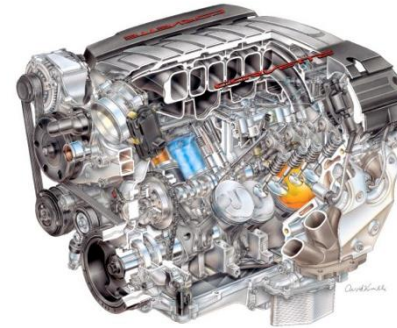
Discriminator network



Module 4: AI to Enhance combustion modeling and predictions

Challenges in Combustion Modeling and Simulations

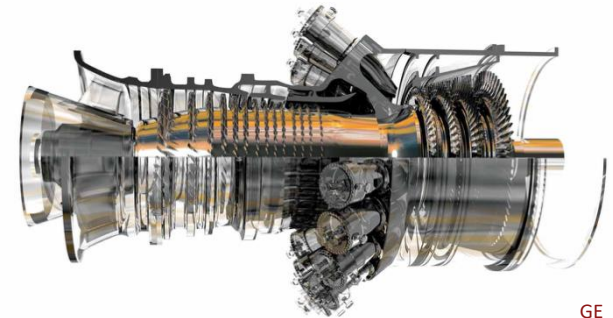
- Important physical phenomenon for power generation and transportation
 - 85% of US energy consumption
- Interesting multi-physics problem
 - Turbulence: *large flow rates*
 - Chemical reaction: *conversion of chemical energy from hydrocarbon fuels*
- Challenging for numerical simulations
 - Wide range of scales: $O(10^0 \sim 10^{-5})$ m, $O(10^{-3} \sim 10^{-10})$ s
 - Large number of species: *50-250 species* **



GM



Blue Origin



GE

Computational modeling of turbulent combustion

LES turbulent combustion models

✓ many species

Manifold-based
combustion models

$$\phi \approx \hat{\phi}^m(\psi^m)$$

- FPV, FPI
- DRG, GPS
- QSSA, PE, RCCE
- ...

✓ small scales

Turbulence-chemistry
closure models
for filtered LES equations

- Presumed-pdf
- Transport-pdf
- Thickened-flame
- ...

Manifold-based combustion models

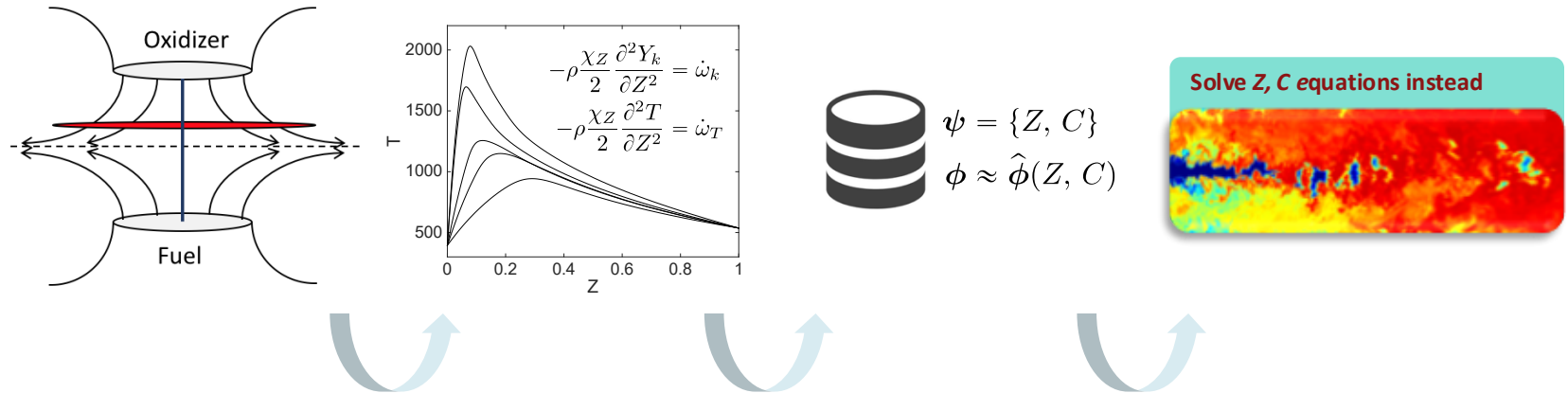
- Topology-based models

- Reduction-based models

Manifold-based combustion models

- **Topology-based models**

- Example: flamelet/progress-variable model (FPV)



Z : mixture fraction, C : progress variable

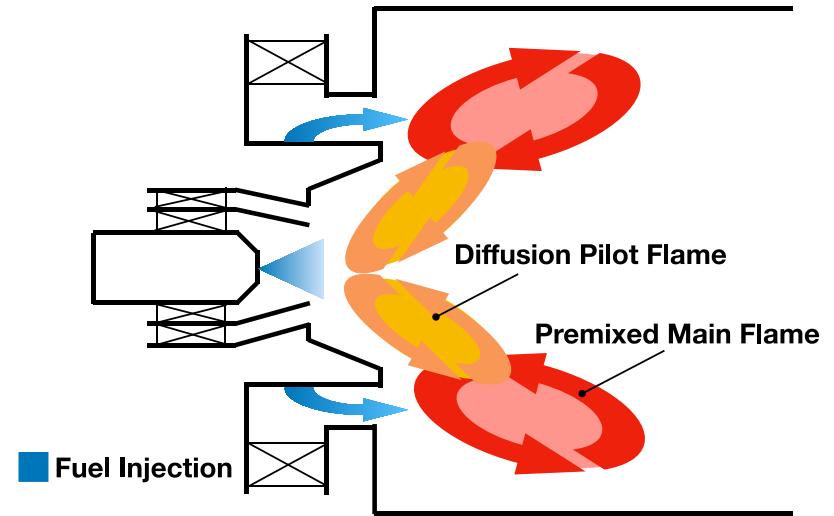
Pierce, C.D. & Moin, P. (2004). *J. Fluid Mech.*, 504

Ihme, M., & Pitsch, H. (2008). *Combust. Flame*, 155

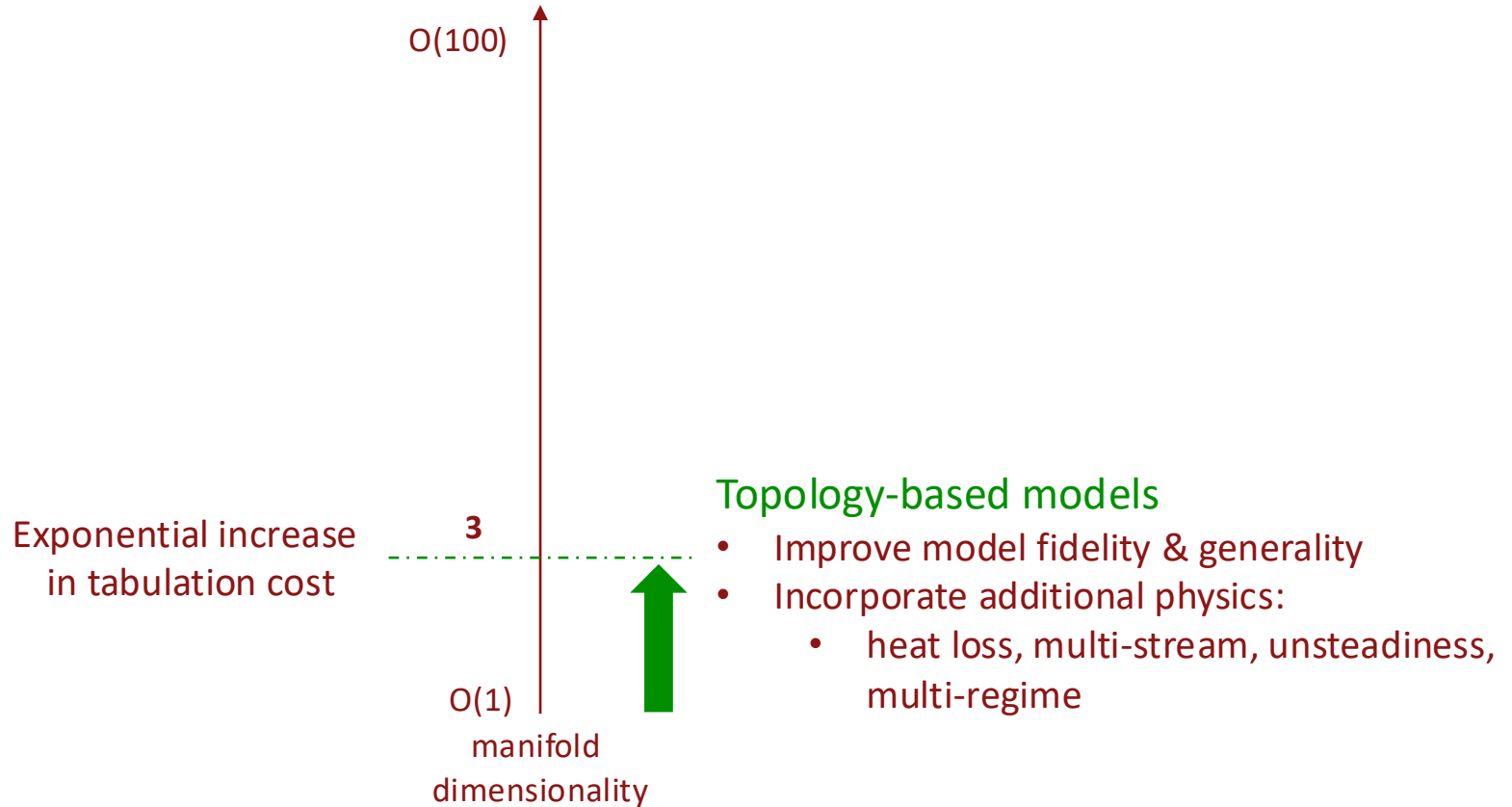
Manifold-based combustion models

- **Topology-based models**

- Construct from idealized canonical flame configurations
- **Low** manifold dimensionality (2, 3)
- **No** algebraic expression for $\hat{\phi}(\psi)$
- **Strong** dependency on combustion regime and flame structure
- **Lower** computational cost
- Require **pre-computation** and **tabulation**
- Examples: flamelet-type models (FPV, FPI, FGM, etc.)



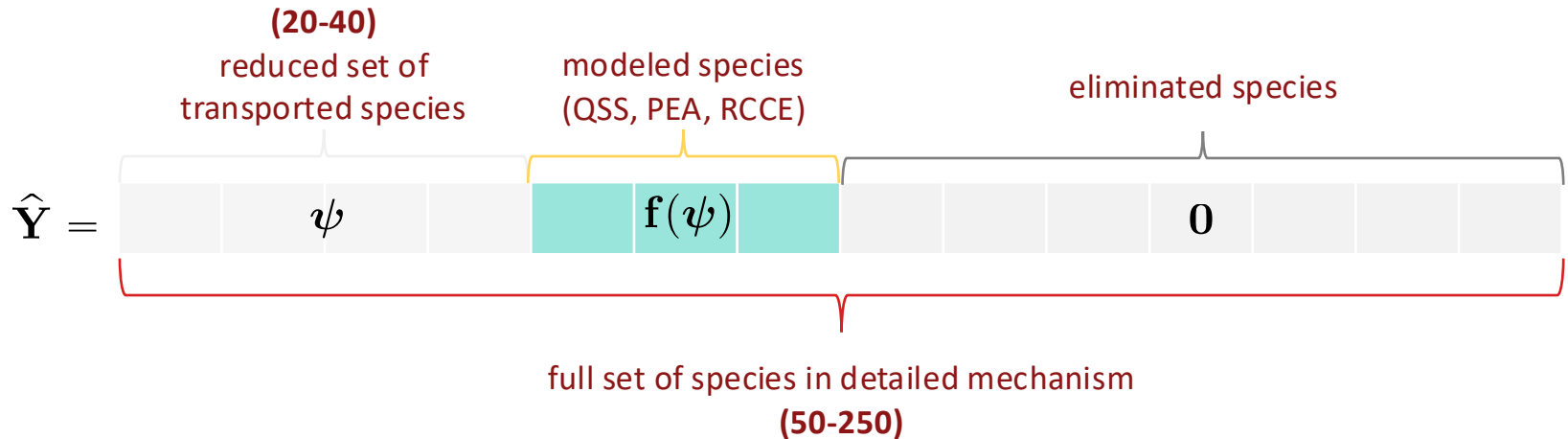
Manifold-based combustion models



Manifold-based combustion models

- Reduction-based models

- Example: mechanism reduction



Manifold-based combustion models

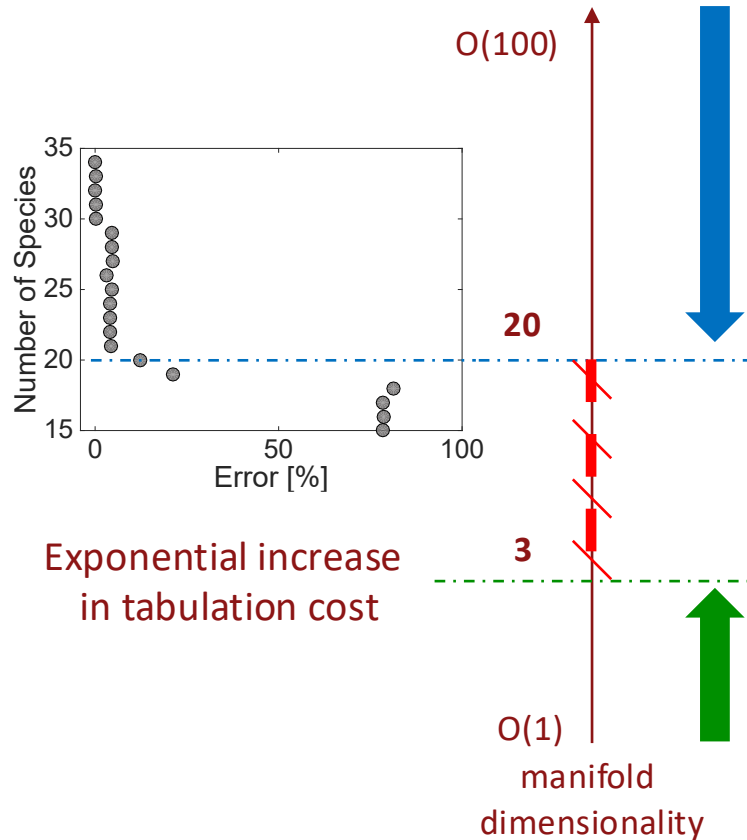
Topology-based models

- Construct from idealized canonical flame configurations
- **Low** manifold dimensionality (2, 3)
- **No** algebraic expression for $\hat{\phi}(\psi)$
- **Strong** dependency on combustion regime and flame structure
- **Lower** computational cost
- Require **pre-computation** and **tabulation**
- Examples: flamelet-type models (FPV, FPI, ...)

Reduction-based models

- Incremental reduction via physical assumptions/simplification
- **High** manifold dimensionality (20-40)
- **Simple** algebraic expression for $\hat{\phi}(\psi)$
- **Weak** dependency on combustion regime and flame structure
- **Higher** model fidelity
- **On-the-fly** evaluation of modeled species
- Examples: DRG, PFA, QSS, PE, RCCE

Manifold-based combustion models



Reduction-based models

- Reduce computational cost
- More effective identification of removable species
- Lumping, fine tuning, etc.

Topology-based models

- Improve model fidelity & generality
- Incorporate additional physics:
 - heat loss, multi-stream, unsteadiness, multi-regime

ML Opportunities

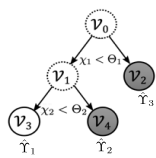
- Large-eddy simulations enable predictions of complex combustion processes through solution of filtered conservation equations:

$$\begin{aligned}\partial_t \bar{\rho} + \nabla \cdot (\bar{\rho} \tilde{\mathbf{u}}) &= 0 \\ \partial_t (\bar{\rho} \tilde{\mathbf{u}}) + \nabla \cdot (\bar{\rho} \tilde{\mathbf{u}} \tilde{\mathbf{u}}) &= -\nabla \cdot (\bar{p} \mathbf{I}) + \nabla \cdot (\bar{\boldsymbol{\tau}}_v + \boldsymbol{\tau}^{sgs}) \\ \partial_t (\bar{\rho} \tilde{e}_t) + \nabla \cdot [\tilde{\mathbf{u}} (\bar{\rho} \tilde{e}_t + \bar{p})] &= -\nabla \cdot (\bar{\mathbf{q}}_v + \mathbf{q}^{sgs}) + \nabla \cdot [(\bar{\boldsymbol{\tau}}_v + \boldsymbol{\tau}^{sgs}) \cdot \tilde{\mathbf{u}}] \\ \partial_t (\bar{\rho} \tilde{Y}_k) + \nabla \cdot (\bar{\rho} \tilde{\mathbf{u}} \tilde{Y}_k) &= -\nabla \cdot (\bar{\mathbf{j}}_v + \mathbf{j}^{sgs}) + \bar{\dot{\omega}}_k \quad \text{where } k = 1, 2, \dots, N_s - 1\end{aligned}$$

- High computational costs arises from:
 - Many species
 - Multiple scales and chemical stiffness
 - Closure models for turbulence chemistry interaction and turbulent transport
- Cost-effective ML methods can help address these challenges

ML applications for turbulent combustion modeling

Discovering combustion closure models



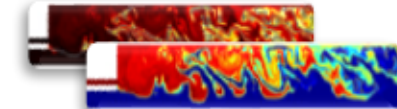
τ_{11}^{sgs}	...	τ_{33}^{sgs}
2.3	...	8.4
...
6.2	...	6.4



Regressing Closure Terms

Combustion model selection

Input: T, Z, \dots

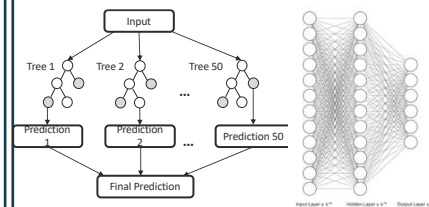
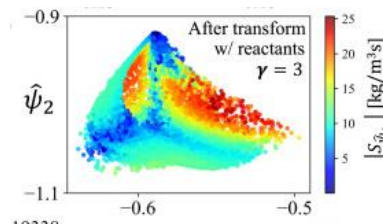


Output: 3 Combustion Submodel Classes

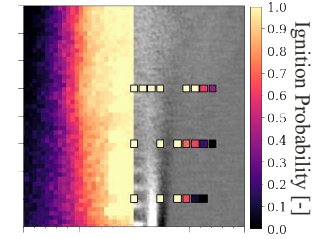


Combustion submodel assignment

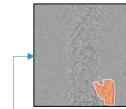
Combustion manifold parameterization



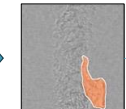
Supplementing Sparse Data with Knowledge



Input: Hot and cold mixtures at a given timestep



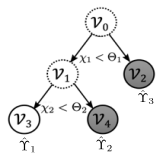
Output: Hot and cold mixtures at a next timestep



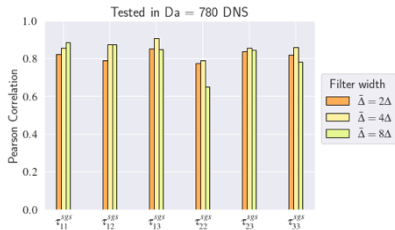
Reduced-order spatio-temporal modeling

ML applications for turbulent combustion modeling

Discovering combustion closure models



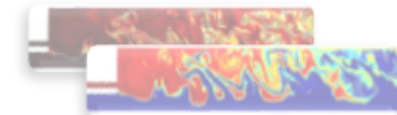
τ_{11}^{sgs}	...	τ_{33}^{sgs}
2.3	...	8.4
...
6.2	...	6.4



Regressing Closure Terms

Combustion model selection

Input: T, Z, \dots



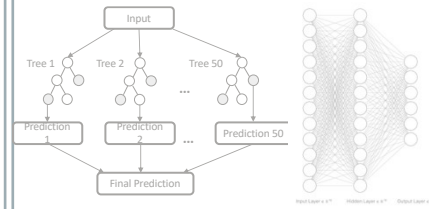
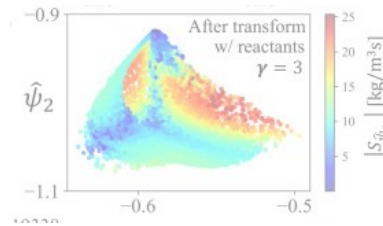
ML

Output: 3 Combustion Submodel Classes

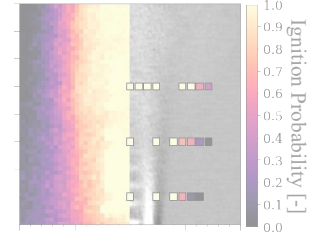


Combustion submodel assignment

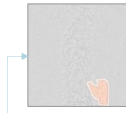
Combustion manifold parameterization



Supplementing Sparse Data with Knowledge



Input: Hot and cold mixtures at a given timestep



Output: Hot and cold mixtures at a next timestep



ML

Reduced-order spatio-temporal modeling

ML Opportunities

- Large-eddy simulations enable predictions of complex combustion processes through solution of filtered conservation equations:

$$\partial_t \bar{\rho} + \nabla \cdot (\bar{\rho} \tilde{\mathbf{u}}) = 0$$

$$\partial_t (\bar{\rho} \tilde{\mathbf{u}}) + \nabla \cdot (\bar{\rho} \tilde{\mathbf{u}} \tilde{\mathbf{u}}) = -\nabla \cdot (\bar{p} \mathbf{I}) + \nabla \cdot (\bar{\boldsymbol{\tau}}_v + \boldsymbol{\tau}^{sgs})$$

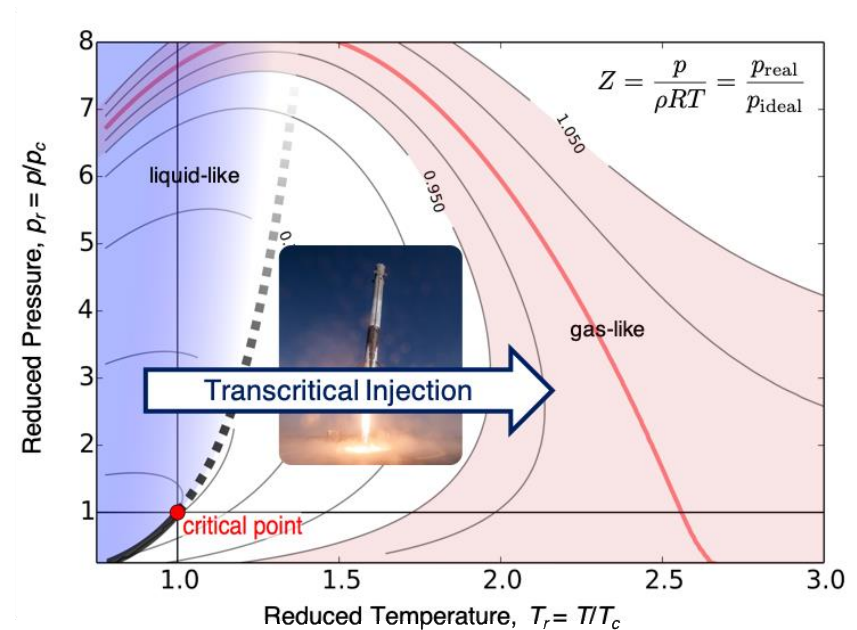
$$\partial_t (\bar{\rho} \tilde{e}_t) + \nabla \cdot [\tilde{\mathbf{u}} (\bar{\rho} \tilde{e}_t + \bar{p})] = -\nabla \cdot (\bar{\mathbf{q}}_v + \mathbf{q}^{sgs}) + \nabla \cdot [(\bar{\boldsymbol{\tau}}_v + \boldsymbol{\tau}^{sgs}) \cdot \tilde{\mathbf{u}}]$$

$$\partial_t (\bar{\rho} \tilde{Y}_k) + \nabla \cdot (\bar{\rho} \tilde{\mathbf{u}} \tilde{Y}_k) = -\nabla \cdot (\bar{\mathbf{j}}_v + \mathbf{j}^{sgs}) + \bar{\omega}_k \quad \text{where } k = 1, 2, \dots, N_s - 1$$

- High computational costs arises from:
 - Many species
 - Multiple scales and chemical stiffness
 - Closure models for turbulence chemistry interaction and turbulent transport
- ML methods can help address these challenges

Problem formulation

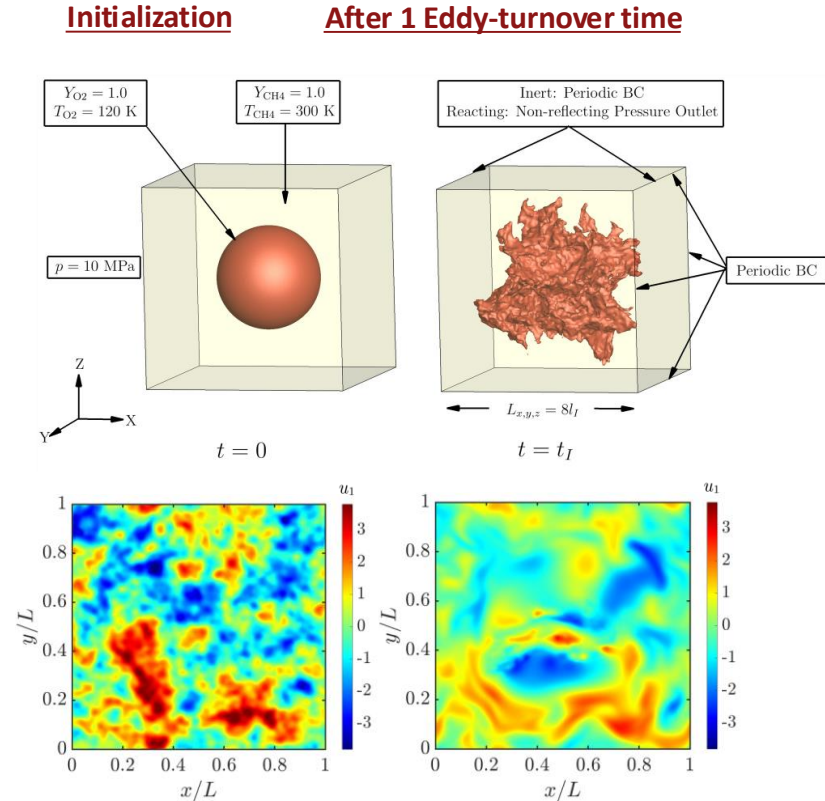
- Develop SGS closure models for high-pressure combustion environments using ML methods?
- Can we extract physical insight from ML methods?
 - LOX-GCH4 engines operate under transcritical conditions
 - SGS Models traditionally been developed for incompressible flows



Problem formulation

• Computational Setup

- DNS: 128^3 domain and $\Delta = O(\eta_K)$
- Diffuse Interface Method
- Peng-Robinson Cubic EoS
- CH₄/O₂ chemistry: 5-species
- Initial conditions: von Karman/Pao spectrum and decaying turbulence
- Initial scalar flowfield initialized with 1D counterflow diffusion flame



Ma et al., J. Comput. Phys. (2017)

Peng, Robinson, Ind.Eng.Chem.Fundam. (1976)

Franzelli et al., Combust. Flame (2012)

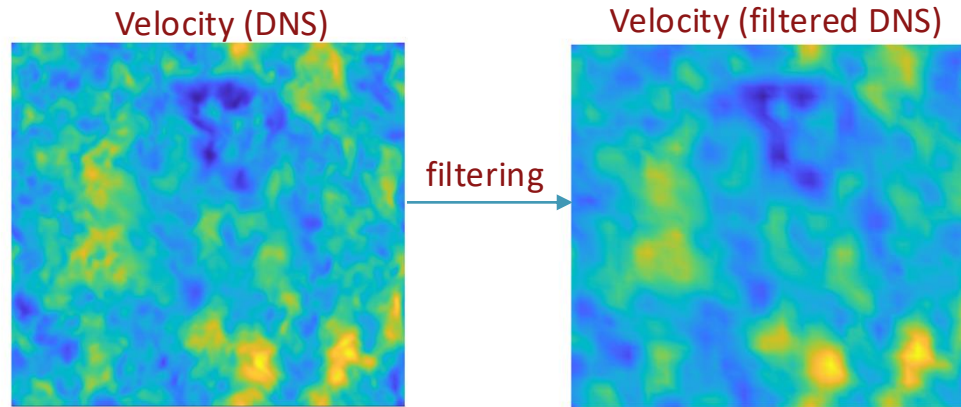
Saad et al. AIAA J. (2017)

Problem formulation

- Apply box-filtering on DNS:

$$\bar{\phi}(x) = \int_V \phi(y) H(x - y; \bar{\Delta}) dy$$

- Extract exact term: $\tau_{ij}^{sgs} = \bar{\rho}(u_i \tilde{u}_j - \tilde{u}_i \tilde{u}_j)$



Regression of SGS closure terms

- ML can help improve cost-effective large-eddy simulations (via closure modeling)
- Hypothesis: $\tau_{ij}^{sgs} = \bar{\rho}(u_i \tilde{u}_j - \tilde{u}_i \tilde{u}_j) = f(\tilde{u}_i, \partial_i \tilde{u}_j, \partial_{ij}^2 \tilde{u}_k)$
- Regression Task

Input: Tabular representation of velocity and derivatives

\tilde{u}_1	$\frac{\partial \tilde{u}_1}{\partial x_1}$	$\frac{\partial^2 \tilde{u}_1}{\partial x_1^2}$...	$\frac{\partial^2 \tilde{u}_3}{\partial x_3^2}$
3.1	4.1	5.9	...	2.6
...
5.3	5.8	9.7	...	9.3

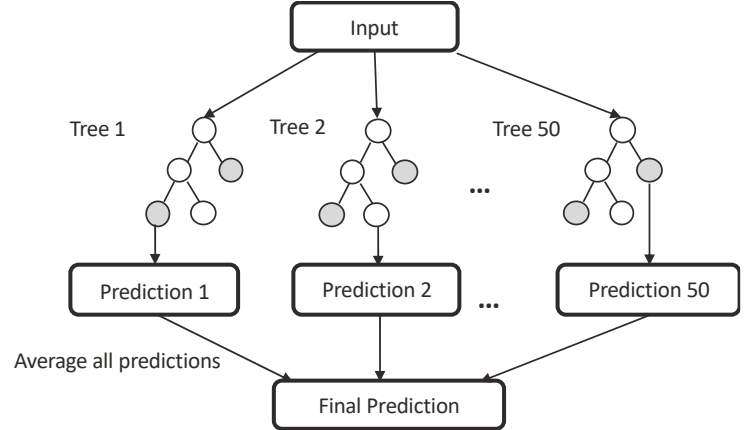


Output:
Corresponding unclosed terms

τ_{11}^{sgs}	...	τ_{33}^{sgs}
2.3	...	8.4
...
6.2	...	6.4

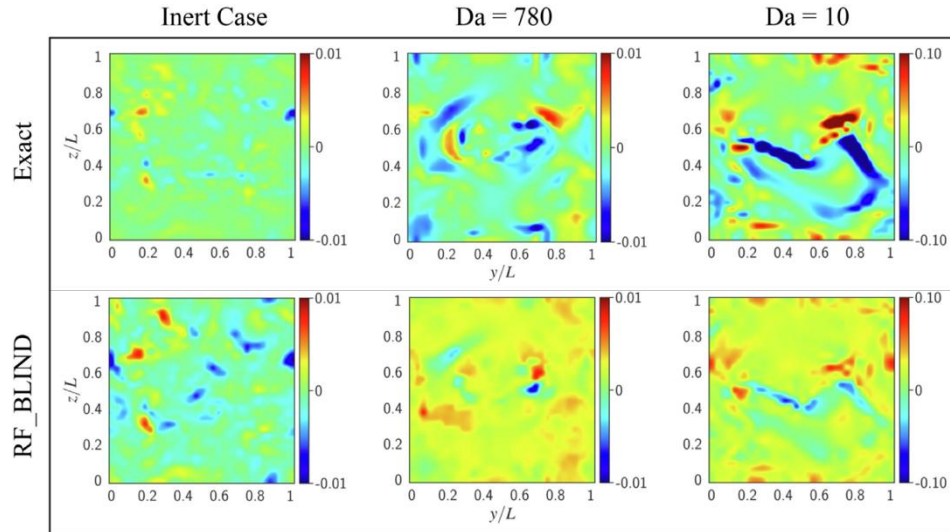
Regression of SGS closure terms

- Hypothesis: $\tau_{ij}^{sgs} = \bar{\rho}(\widetilde{u_i u_j} - \widetilde{u_i} \widetilde{u_j}) = f(\widetilde{u_i}, \partial_i \widetilde{u_j}, \partial_{ij}^2 \widetilde{u_k})$
- Data Processing
 - Favre-filter and downsample DNS data
 - Convert to tabular format
- ML Model
 - Random Forest – Ensemble of Classification and Regression Trees (CaRTs)
 - Developed ML model on time of $t = \tau_{eddy}$
 - Tested model on unseen data at $t = 1.5 \tau_{eddy}$

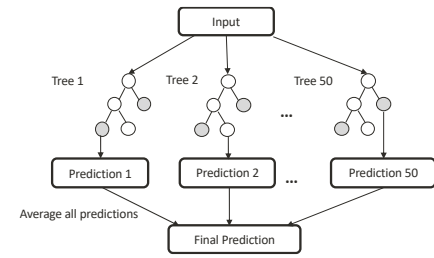


Regression of SGS closure terms

- Predicting τ_{12}^{sgs} [m²s⁻²] at filter width $\bar{\Delta}/\Delta = 4$
 - Captures flow field structures
 - Noticeable errors in magnitude
 - Typical performance for tabular ML: Lack of structural information



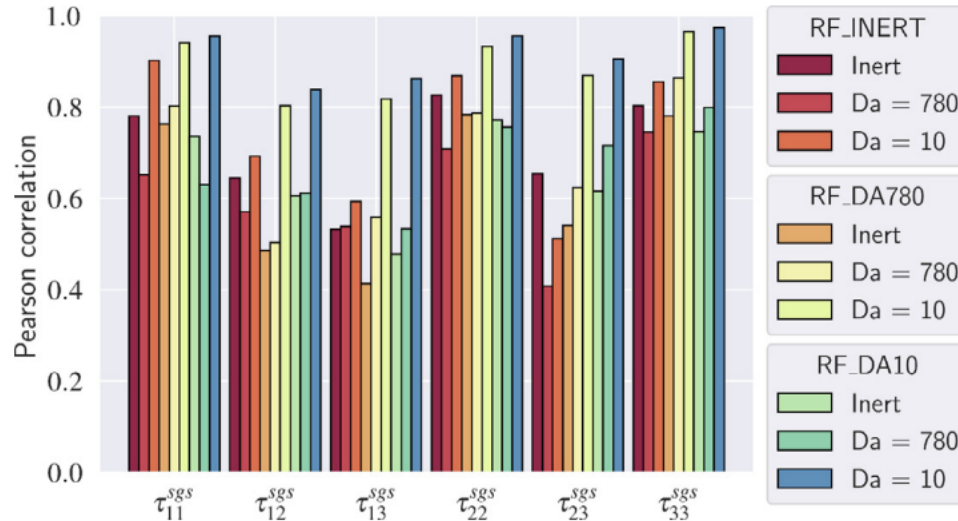
Regression of SGS closure terms



- Physical interpretation

- Pearson correlations: measures linear correlation between two sets of data

$$\rho_{\phi,\psi} = \frac{\mathbb{E}[(\phi - \mu_{\phi})(\psi - \mu_{\psi})]}{\sigma_{\phi}\sigma_{\psi}} \in [-1,1]$$

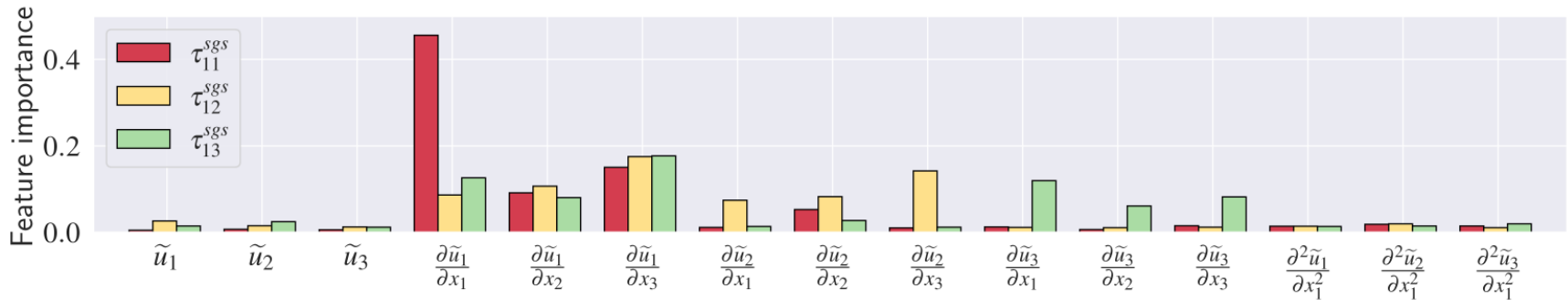
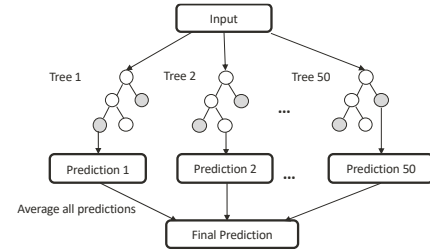


Regression of SGS closure terms

- Physical interpretation

- Pearson correlations
- Feature importance scores

- Integrated gradients technique: $i_G = (x_i - x'_i) \int_{\alpha=0}^1 \partial_{x_i} F(x'_i + \alpha(x_i - x'_i)) d\alpha$
- Mean decrease impurity, Gini index: $i_G(V) = \sum_{k=1}^K P(\psi_k|V)(1 - P(\psi_k|V))$
- Maximal Information Coefficient (MIC)



Sundararajan, Taly, Yan, PMLR 70:3319-3328, 2017
 Gini, Mutabilità: Tipogr. di P. Cuppini, 1912.

Sparse symbolic regression

- Objective: obtain physical insight by discovering mathematical model from data
- Approaches
 - Sparse regression
 - Sparse identification of nonlinear dynamics (SINDy)
 - Least absolute shrinkage and selection operator (LASSO)
- General concept:
 - Consider sample data: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$; assume linear model $y \sim \beta_0 + \beta_1 x$
$$\min_{\beta_0, \beta} \left\{ \sum_{i=1}^N (y_i - \beta_0 - x_i^T \beta)^2 \right\} \text{ s.t. } |\beta| \leq \epsilon$$

Sparse symbolic regression for model discovery

Represent SGS model as linear regression problem: $\mathbf{X}\boldsymbol{\beta} = \mathbf{f}$

- with hypothesis set $\mathbf{X} = \{\tilde{u}_i; \partial_j \tilde{u}_i; \partial_{jk}^2 \tilde{u}_i\}$
- weights: $\boldsymbol{\beta} = [\beta_0 \beta_1 \dots \beta_m]^T$
- Function: $\mathbf{f} = \tau_{ij}^{SGS}$

Introduce nonlinearities: $G^d(\mathbf{X})\boldsymbol{\beta} = \mathbf{f}$

$$G^d(\mathbf{X}) = \begin{bmatrix} 1 & X_{11} & X_{12} & \dots & X_{1n} & X_{11}^2 & X_{11}X_{12} & \dots & X_{1n}^d \\ 1 & X_{21} & X_{22} & \dots & X_{2n} & X_{21}^2 & X_{21}X_{22} & \dots & X_{2n}^d \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{m1} & X_{m2} & \dots & X_{mn} & X_{m1}^2 & X_{m1}X_{m2} & \dots & X_{mn}^d \end{bmatrix}$$

Add L_1 constraint for sparsity promotion:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{m} \|\mathbf{f} - G^d(\mathbf{X})\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}_1\| \right\}$$

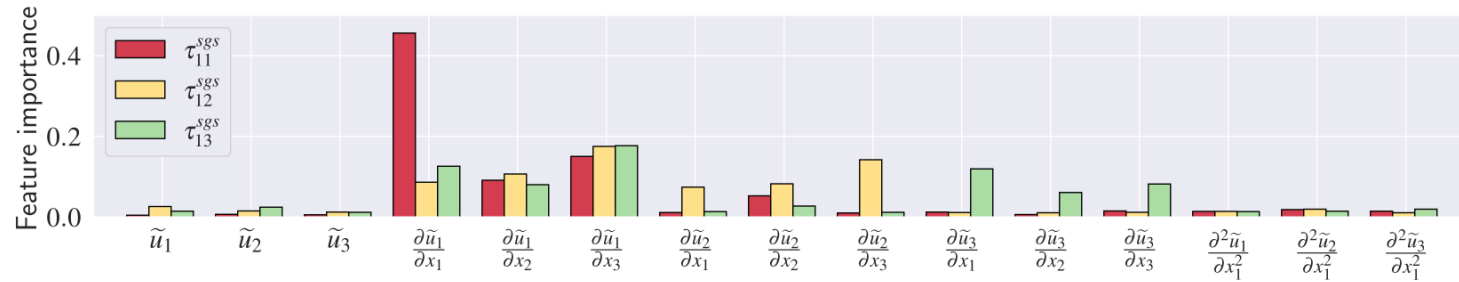
Solve system as optimization problem:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{m} \|\mathbf{f} - G^d(\mathbf{X})\boldsymbol{\beta}\|_2^2 \right\}$$

Find $\boldsymbol{\beta}$ via any optimization scheme gradient descent

Sparse regression for model discovery

- Can be computationally expensive as cost scales with MN^d
- Use feature importance (mean decrease in node impurity) to simplify sparse symbolic regression for model discovery



$$\frac{\tau_{ij}^{sgs}}{\bar{\rho}u'^2} = f_{ij} \left[G^{d=2} \left(\frac{\tilde{u}_i}{u'}, \frac{\bar{\Delta}}{u'} \frac{\partial \tilde{u}_i}{\partial x_j}, \frac{\bar{\Delta}}{u'} \frac{\partial \tilde{u}_j}{\partial x_i}, \frac{\bar{\Delta}^2}{u'} \frac{\partial^2 \tilde{u}_i}{\partial x_j \partial x_k}, \frac{\bar{\Delta}^2}{u'} \frac{\partial^2 \tilde{u}_j}{\partial x_i \partial x_k}, \frac{\bar{\Delta}^2}{u'} \frac{\partial^2 \tilde{u}_k}{\partial x_i \partial x_i} \right) \right]$$



$$\frac{\tau_{ij}^{sgs}}{\bar{\rho}u'^2} = f_{ij} \left[G^{d=2} \left(\frac{\bar{\Delta}}{u'} \frac{\partial \tilde{u}_i}{\partial x_k}, \frac{\bar{\Delta}}{u'} \frac{\partial \tilde{u}_j}{\partial x_k} \right) \right]$$

Sparse regression for model discovery

$$\frac{\tau_{ij}^{sgs}}{\bar{\rho} u'^2} = f_{ij} \left[G^{d=2} \left(\frac{\bar{\Delta}}{u'} \frac{\partial \tilde{u}_i}{\partial x_k}, \frac{\bar{\Delta}}{u'} \frac{\partial \tilde{u}_j}{\partial x_k} \right) \right]$$



Apply sparse symbolic regression

$$\tau_{11}^{sgs} \simeq \bar{\Delta}^{-2} \left[0.116 \frac{\partial \tilde{u}_1}{\partial x_1} \frac{\partial \tilde{u}_1}{\partial x_1} + 0.191 \frac{\partial \tilde{u}_1}{\partial x_2} \frac{\partial \tilde{u}_1}{\partial x_2} + 0.207 \frac{\partial \tilde{u}_1}{\partial x_3} \frac{\partial \tilde{u}_1}{\partial x_3} \right]$$

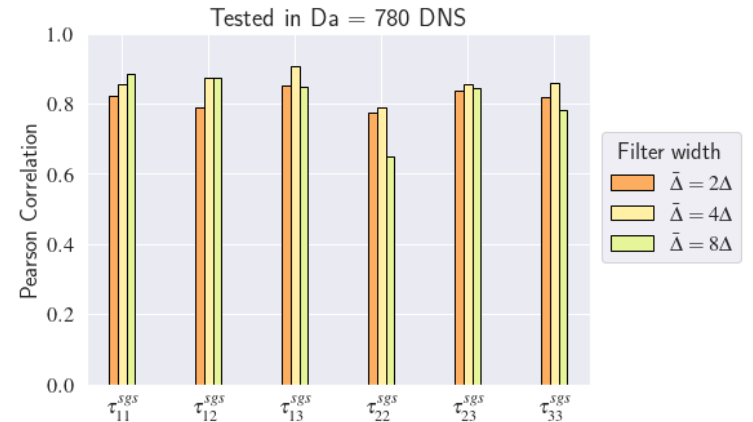
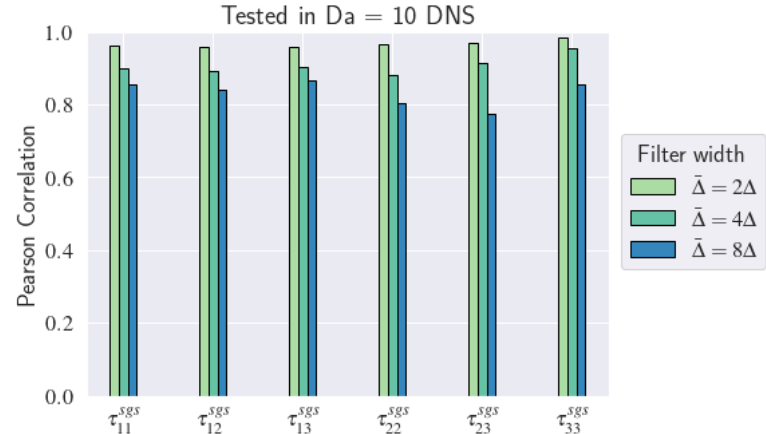
$$\vdots$$

$$\tau_{33}^{sgs} \simeq \bar{\Delta}^{-2} \left[0.251 \frac{\partial \tilde{u}_3}{\partial x_1} \frac{\partial \tilde{u}_3}{\partial x_1} + 0.177 \frac{\partial \tilde{u}_3}{\partial x_2} \frac{\partial \tilde{u}_3}{\partial x_2} + 0.124 \frac{\partial \tilde{u}_3}{\partial x_3} \frac{\partial \tilde{u}_3}{\partial x_3} \right]$$



$$\tau_{ij}^{sgs} \simeq \underbrace{\bar{\rho} C_x}_{\sim 1/6} \bar{\Delta}^{-2} \frac{\partial \tilde{u}_i}{\partial x_k} \frac{\partial \tilde{u}_j}{\partial x_k}$$

**Rediscovered
Clark and Ferziger's
Gradient Model!**



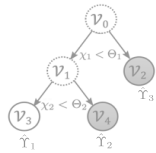
Summary

- Demonstrate ability to extract physically meaningful insight from data
- Random forest are interpretable and can reveal most important inputs
- Feature importance can be used with sparse symbolic regression for model discovery

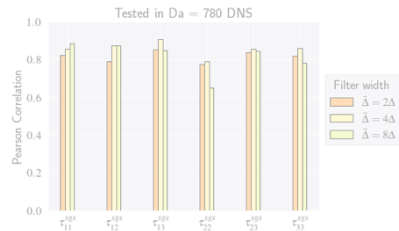
- Opportunities
 - Extension of turbulent closure models
 - Interpretable foundation ML models for combustion closure models

ML applications for turbulent combustion modeling

Discovering combustion closure models



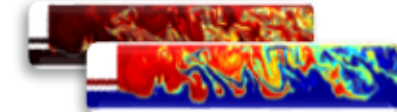
τ_{11}^{sgs}	...	τ_{33}^{sgs}
2.3	...	8.4
...
6.2	...	6.4



Regressing Closure Terms

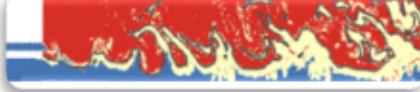
Combustion model selection

Input: T, Z, \dots



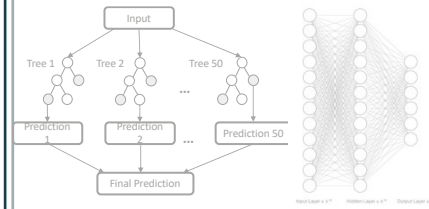
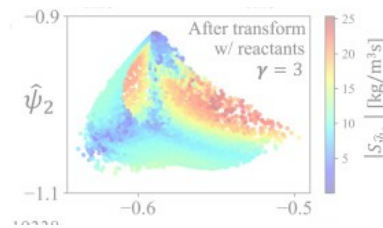
ML

Output: 3 Combustion Submodel Classes

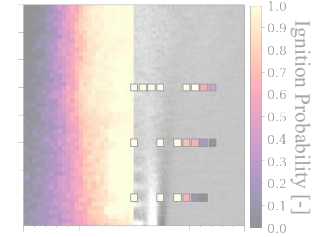


Combustion submodel assignment

Combustion manifold parameterization



Supplementing Sparse Data with Knowledge



Input: Hot and cold mixtures at a given timestep



Output: Hot and cold mixtures at a next timestep



ML

Reduced-order spatio-temporal modeling

ML Opportunities

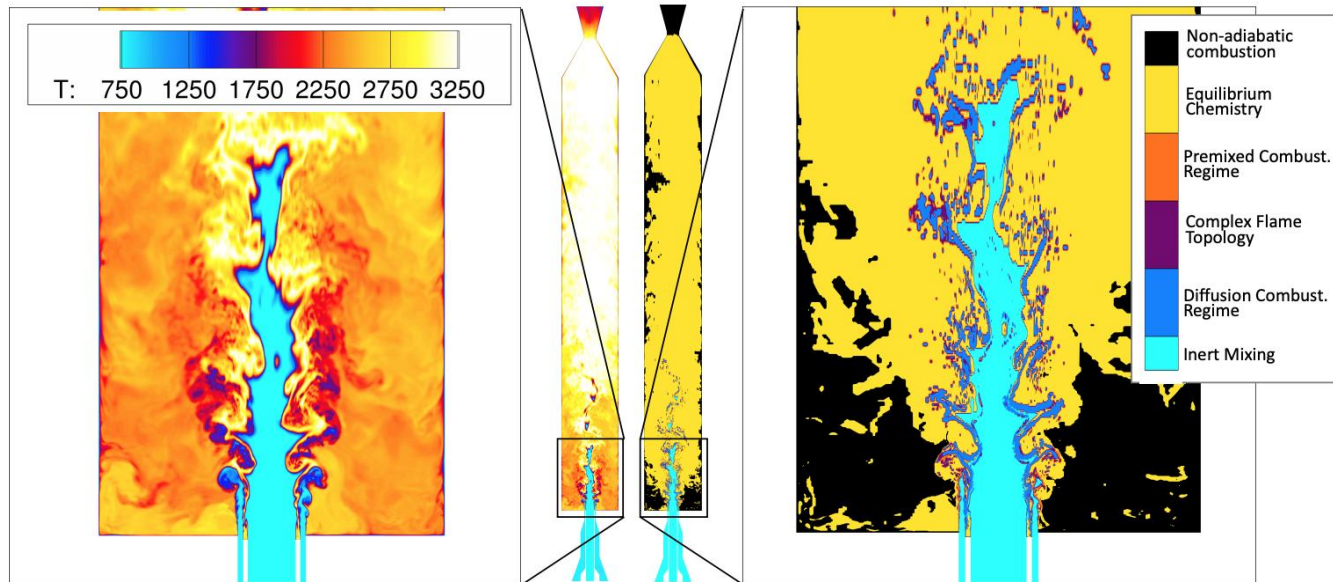
- Large-eddy simulations enable predictions of complex combustion processes through solution of filtered conservation equations:

$$\begin{aligned}\partial_t \bar{\rho} + \nabla \cdot (\bar{\rho} \tilde{\mathbf{u}}) &= 0 \\ \partial_t (\bar{\rho} \tilde{\mathbf{u}}) + \nabla \cdot (\bar{\rho} \tilde{\mathbf{u}} \tilde{\mathbf{u}}) &= -\nabla \cdot (\bar{p} \mathbf{I}) + \nabla \cdot (\bar{\boldsymbol{\tau}}_v + \boldsymbol{\tau}^{sgs}) \\ \partial_t (\bar{\rho} \tilde{e}_t) + \nabla \cdot [\tilde{\mathbf{u}} (\bar{\rho} \tilde{e}_t + \bar{p})] &= -\nabla \cdot (\bar{\mathbf{q}}_v + \mathbf{q}^{sgs}) + \nabla \cdot [(\bar{\boldsymbol{\tau}}_v + \boldsymbol{\tau}^{sgs}) \cdot \tilde{\mathbf{u}}] \\ \partial_t (\bar{\rho} \tilde{Y}_k) + \nabla \cdot (\bar{\rho} \tilde{\mathbf{u}} \tilde{Y}_k) &= -\nabla \cdot (\bar{\mathbf{j}}_v + \mathbf{j}^{sgs}) + \bar{\dot{\omega}}_k \quad \text{where } k = 1, 2, \dots, N_s - 1\end{aligned}$$

- High computational costs arises from:
 - Many species
 - Multiple scales and chemical stiffness
 - Closure models for turbulence chemistry interaction and turbulent transport
- Cost-effective ML methods can help address these challenges

Motivation

- High-fidelity simulations of turbulent reacting flows incur high computational cost
- Selection of turbulent combustion model requires consideration of underlying flow-field representation



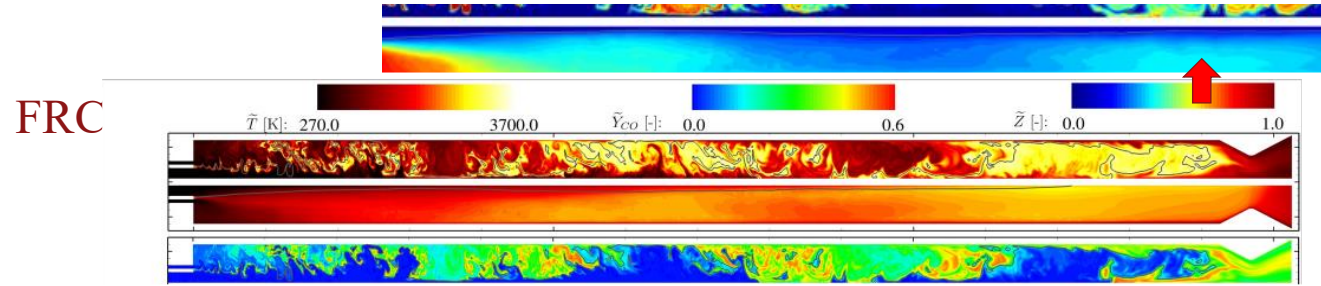
Problem with ML-based regression

- **Physical models** versus **data-driven models**: **conservation laws** versus **complex cross-correlations**
- Data-driven models may violate physics during extrapolation tasks
- Data driven models are prone to numerical instability

- **Solution:** Use **data-driven method** to **assist** the selection of low-fidelity **physics-based model** through **classification**

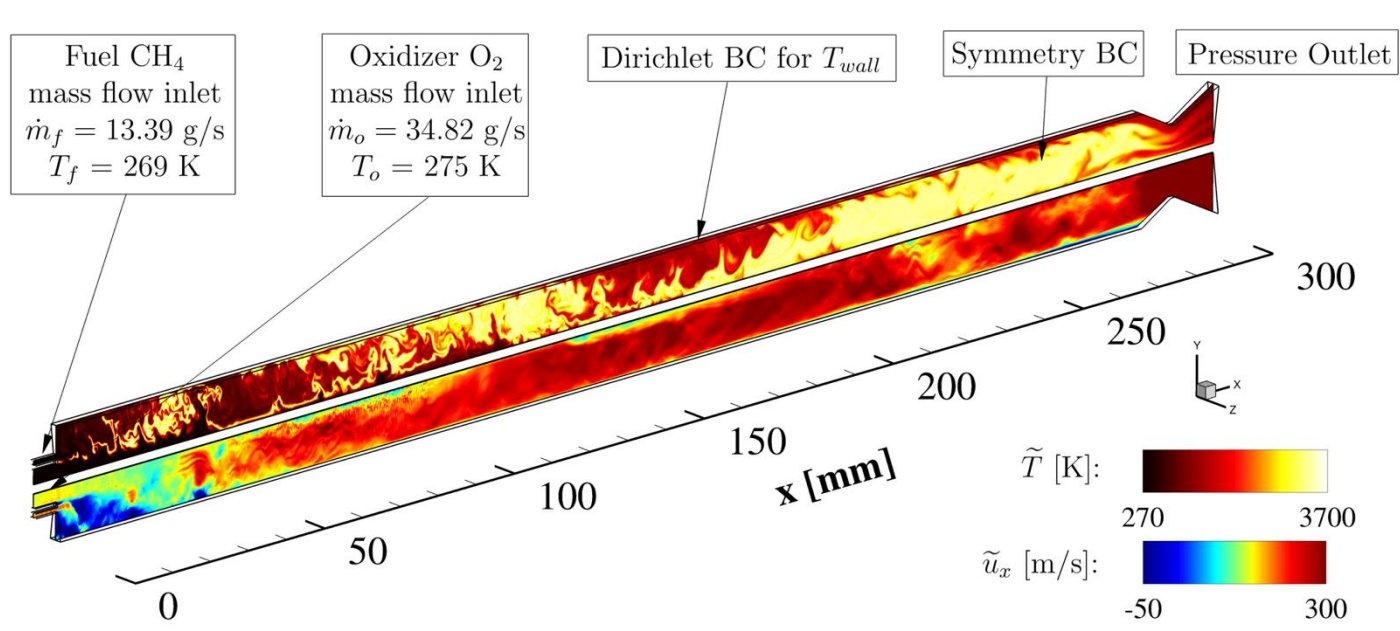
Objective

- Finite rate chemistry (FRC) very accurate but computationally expensive
- Flamelet Progress Variable (FPV) model cannot capture thermal boundary layers
- ➔ Use ML to improve on FPV simulations at a lower cost than FRC



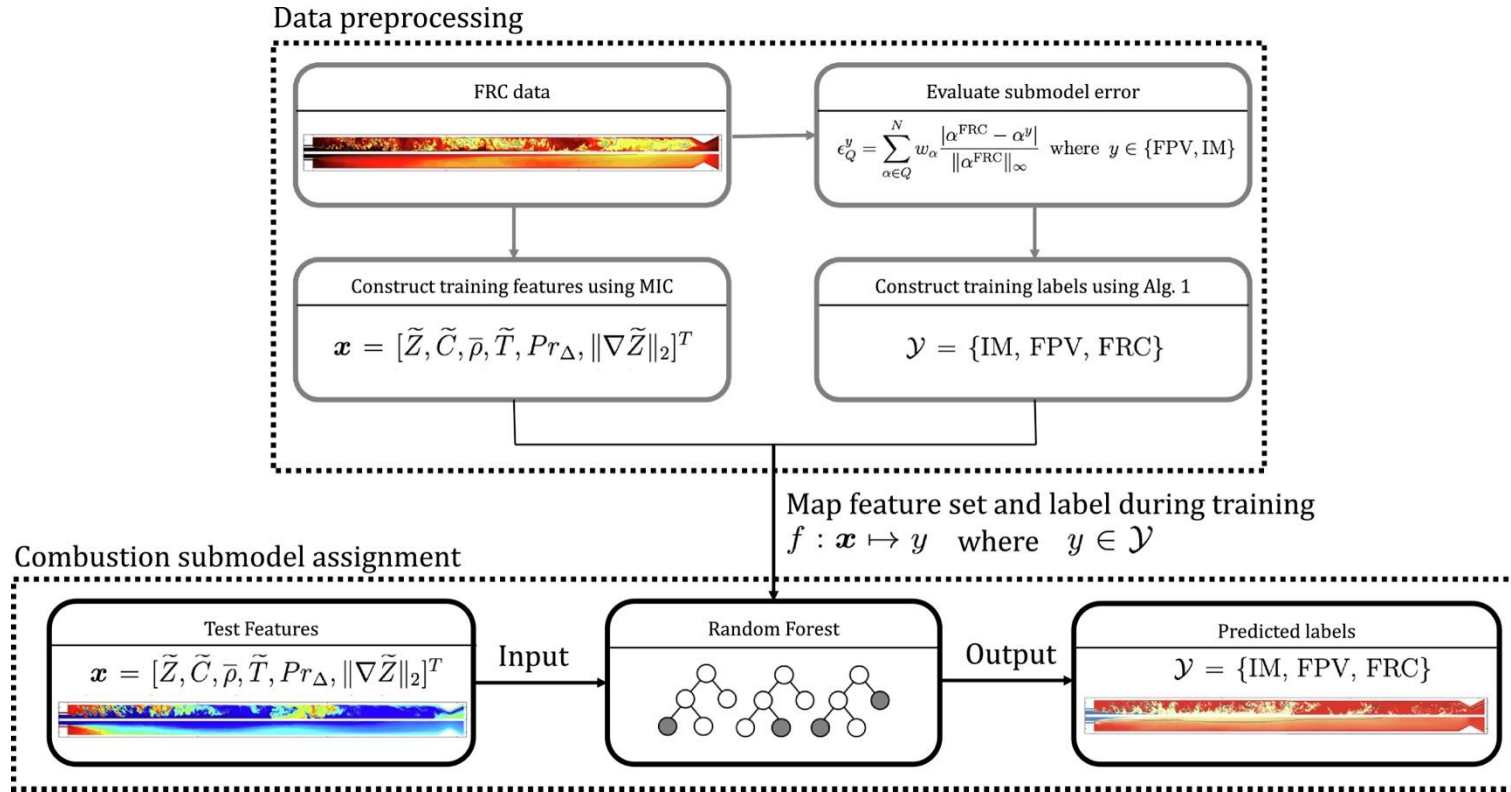
Configuration

- Experiment: TUM GOX/GCH4 single element rocket combustor
- 2×10^5 cells, axisymmetric domain, with minimum cell size of $30 \mu\text{m}$



Silvestri, Celano, Haidn, and Knab, *EUCASS* (2015)

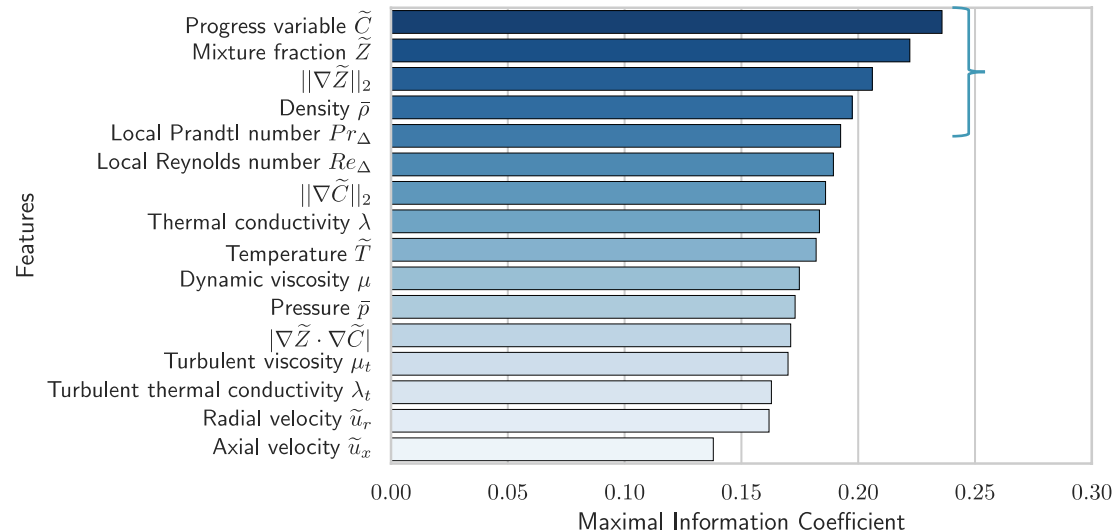
Data-assisted LES



Feature/input selection

- Use Maximal Information Coefficient to select most relevant input
- MIC is a correlation measure for nonlinear data, similar to R2 measure

MIC relating features with $\epsilon_{\{T,CO\}}^{FPV}$



Constructing training labels/output

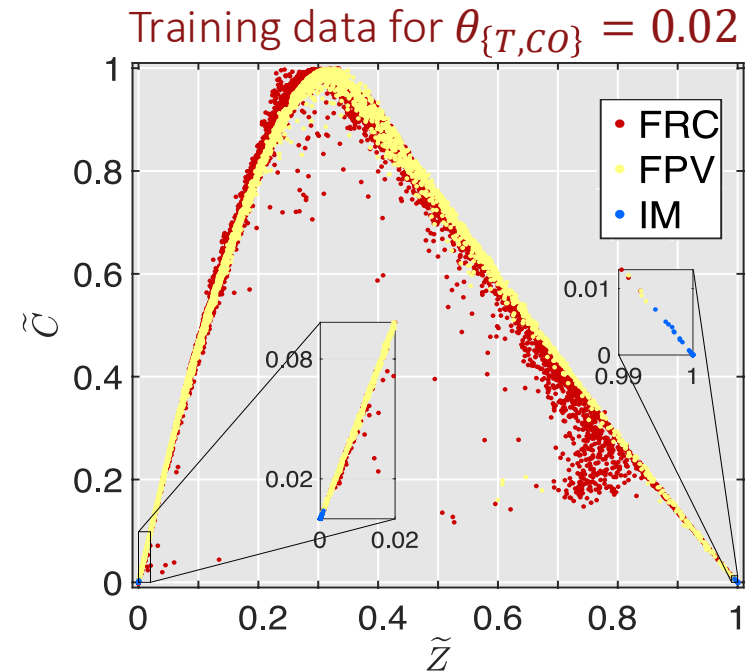
$$\theta_{\{T, CO\}} = 0.02$$

- Error between different submodels for QOI: $\alpha = \{\tilde{T}, \tilde{Y}_{CO}\}$

$$\epsilon_Q^y = \sum_{\alpha \in Q} w_\alpha \frac{|\alpha^{\text{FRC}} - \alpha^y|}{\|\alpha^{\text{FRC}}\|_\infty} \quad \text{with } y \in \{\text{FPV}, \text{IM}\}$$

User defined threshold

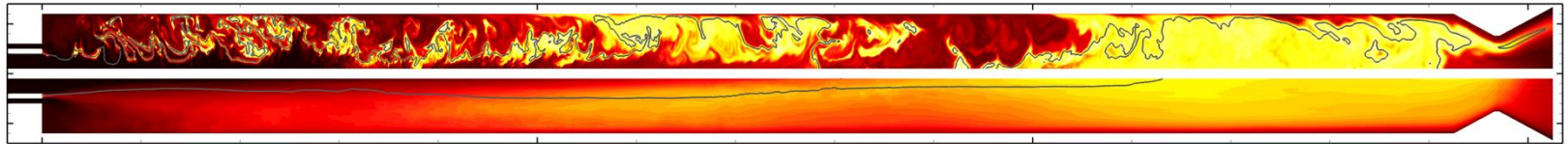
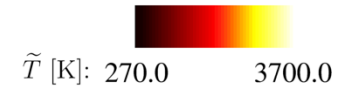
```
if  $\epsilon_Q^{\text{IM}} < \theta_Q^{\text{IM}}$  then
  | use inert mixing (IM)
else if  $\epsilon_Q^{\text{FPV}} < \theta_Q^{\text{FPV}}$  then
  | use tabulated chemistry (FPV)
else
  | use finite-rate chemistry (FRC)
end
```



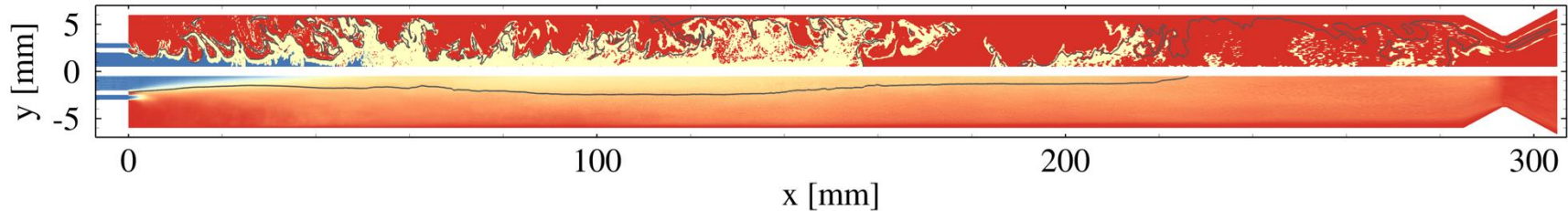
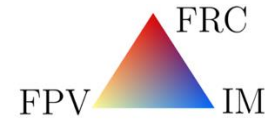
A posteriori results

- Temperature field

$$DA, \theta_{\{T, CO\}} = 0.02$$

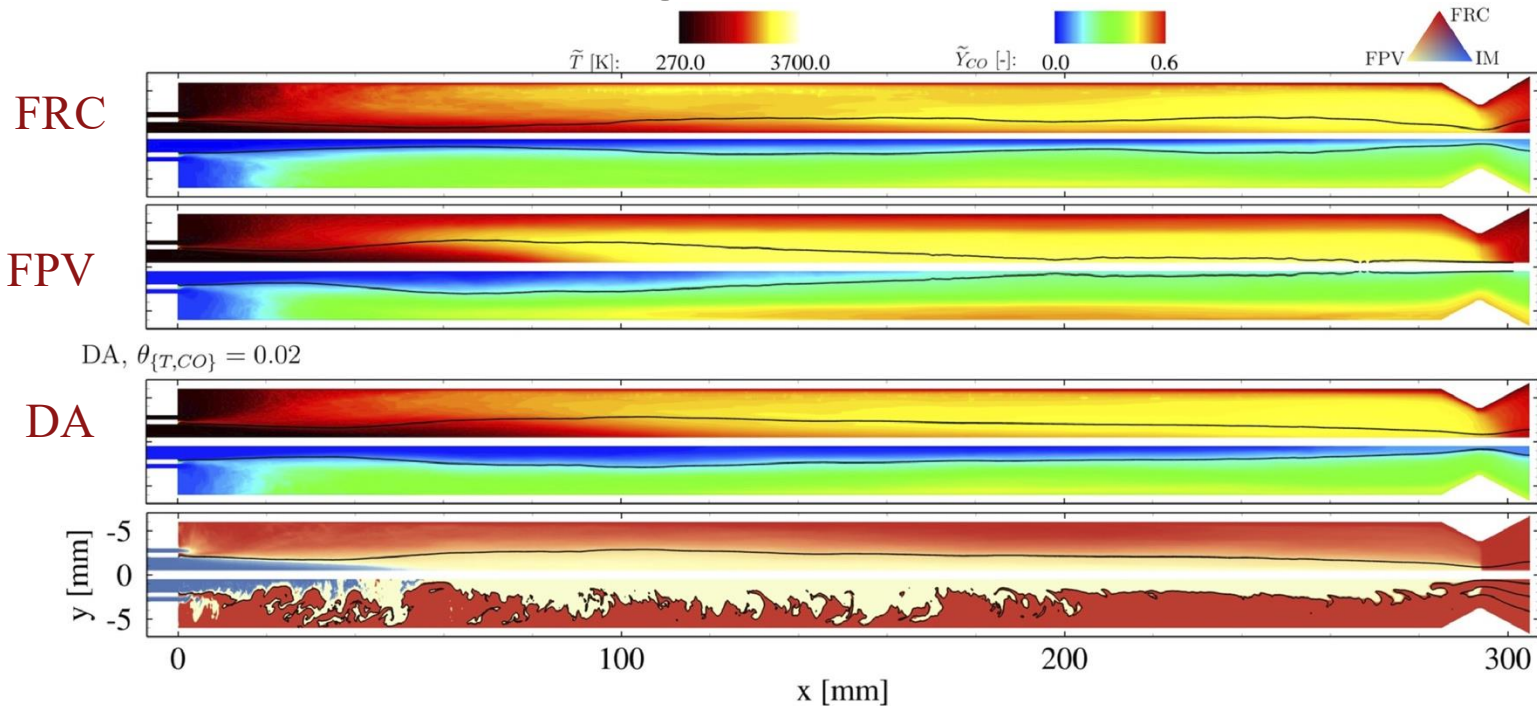


- Combustion model assignment



A posteriori results: Extrapolation task?

- For a modified configuration with 3x inlet mass flow rate
- Method demonstrates ability to **generalize** for different configurations

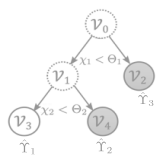


Summary

- ML-based combustion classifier
 - Classification can be used to locally assign combustion models
 - Overall desired fidelity can be controlled during labelling
 - DA simulations outperform monolithic FPV and monolithic FRC simulations in accuracy and cost, respectively
 - Random forest and neural networks demonstrate similar accuracy *a priori*, but show different behavior
 - Neural networks have a more contiguous assignment than random forests *a posteriori*
- Further advances
 - Inclusion of generic physical laws during labeling and training
 - Extend submodel selection to turbulence modeling, and equation of state

ML applications for turbulent combustion modeling

Discovering combustion closure models



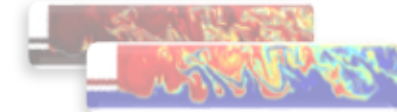
τ_{11}^{sgs}	...	τ_{33}^{sgs}
2.3	...	8.4
...
6.2	...	6.4



Regressing Closure Terms

Combustion model selection

Input: T, Z, \dots

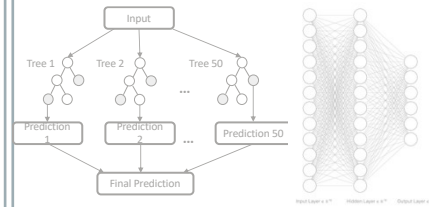
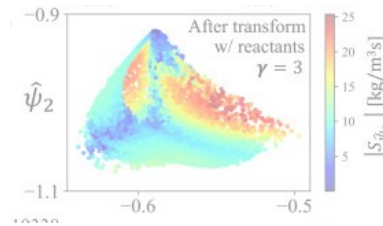


Output: 3 Combustion Submodel Classes

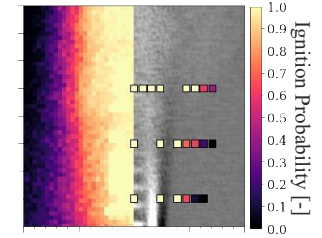


Combustion submodel assignment

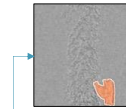
Combustion manifold parameterization



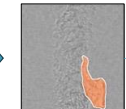
Supplementing Sparse Data with Knowledge



Input: Hot and cold mixtures at a given timestep



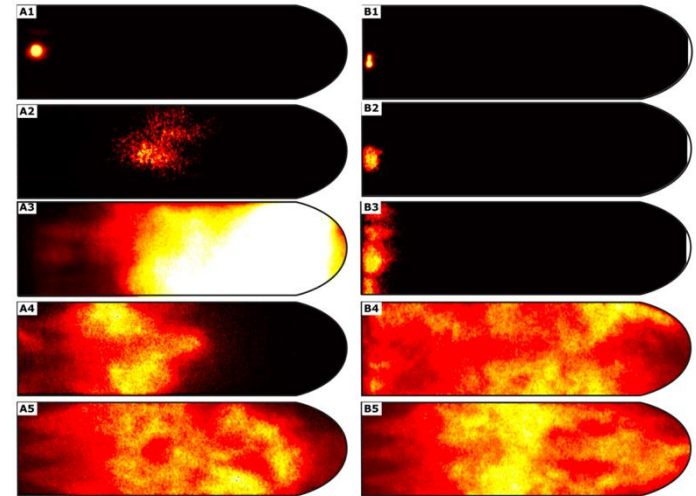
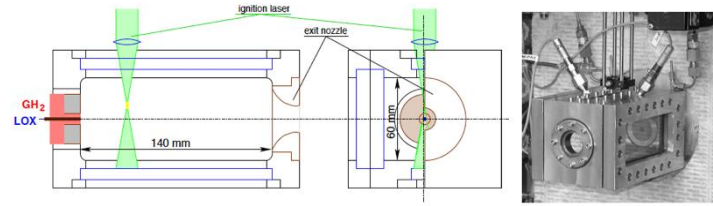
Output: Hot and cold mixtures at a next timestep



Reduced-order spatio-temporal modeling

Background: reliable ignition

- Reliable ignition for space propulsion
- Costly experiments
- Limited measurements



Mastorakos, Prog. Energy Combust. Sci., 35, 2009

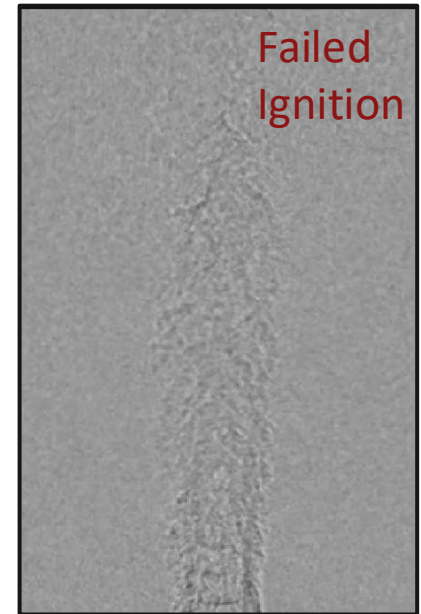
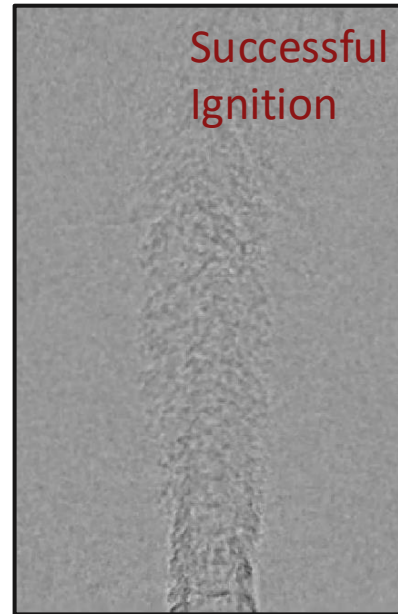
Manfretti: Laser Ignition of an Experimental Cryogenic Reaction and Control Thruster: Pre-Ignition Conditions, JPP, 2014

Schmidt, O. Gurliat, M. Oswald and O.J. Haidn, "Influence of J and We Number on GH2/LOX Ignition Process," ILASS-Europe (2002)

Gurliat, V. Schmidt, O. J. Haidn, M. Oswald, "Ignition of cryogenic H2/LOX sprays." Aerospace Science and Technology 7 (2003) 517-531

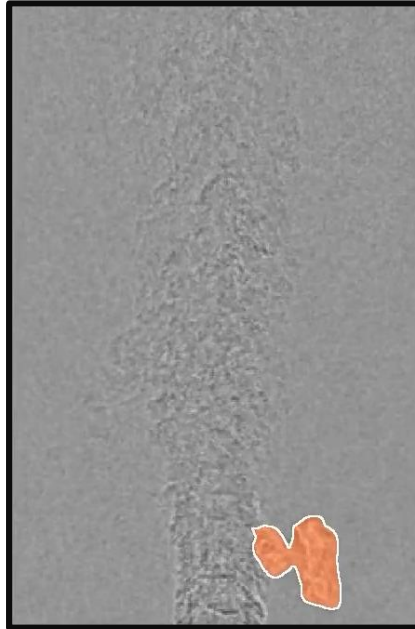
Background: reliable ignition

- Reliable ignition for space propulsion
- Costly experiments
- Limited measurements
- Stochastic ignition
 - Turbulent mixing
 - Variations in laser energy

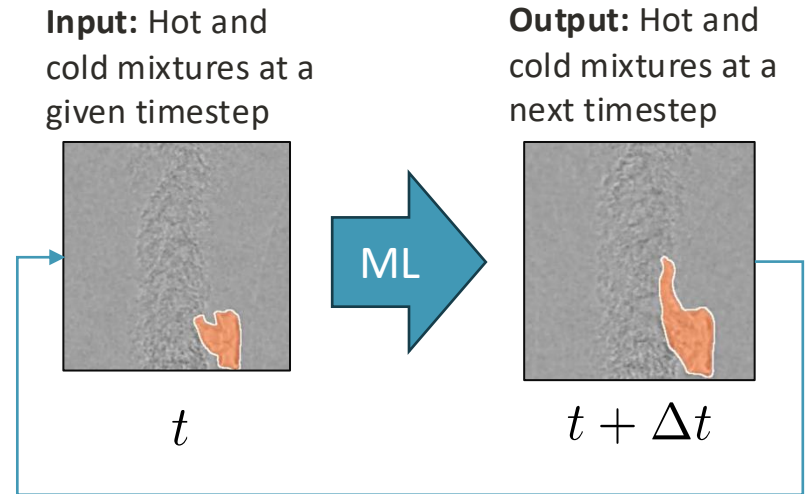


Reduced-order Modeling of Ignition Kernels

Only Interested in Ignition Kernel Behavior



Classification Task



Iterate as needed (autoregressive modeling)

Enables affordable reduced-order modeling of specific flow phenomena

Challenges in data-sparse ML modeling

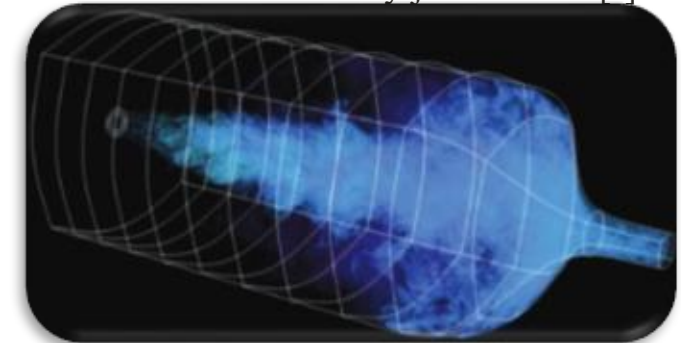
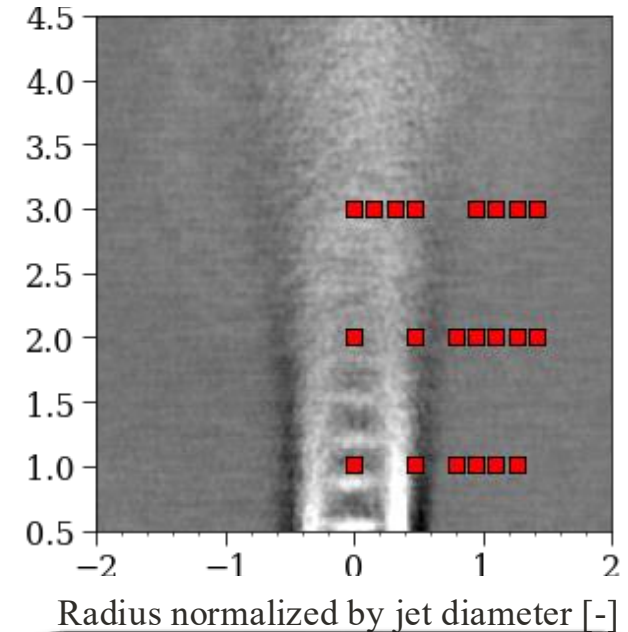
- Question: How can we combine sparse experimental observations and limited simulations to create reduced-order data-driven models

Setup

• Configuration

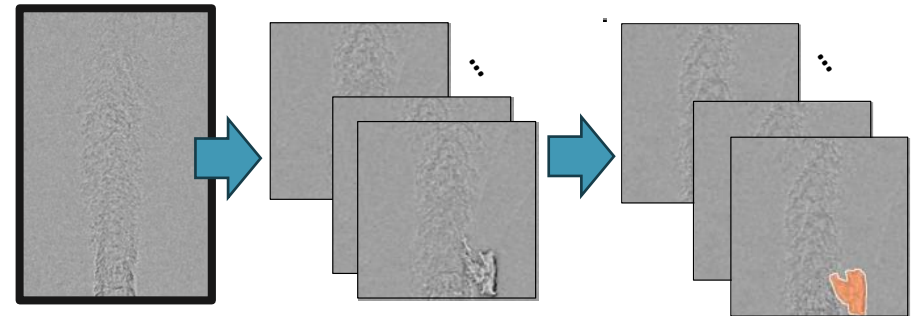
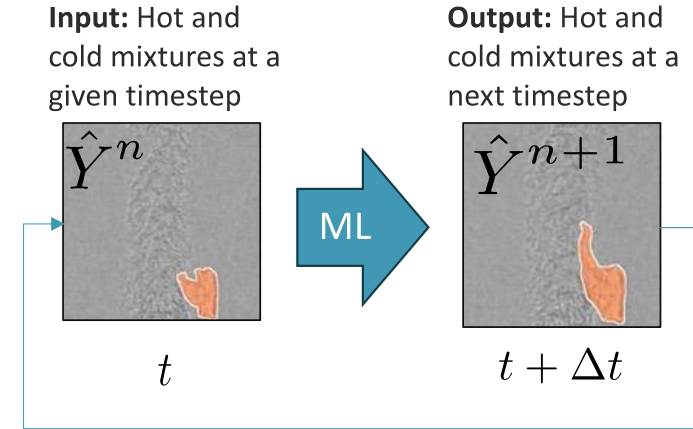
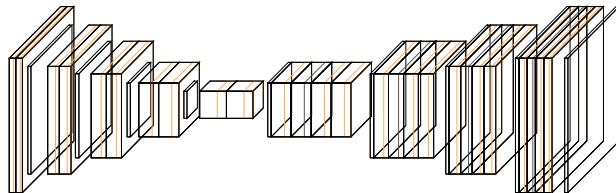
- Gaseous CH₄/O₂ model rocket combustor by Strelau *et al.*
- Laser deposited at ~21 mJ
- Schlieren measurements from 153 ignition tests at 21 laser locations
- Simulation data from 3D inert large-eddy simulation
- $T_o = 242$ K, $T_f = 282$ K and $P = 1.4$ bar
- Co-axial injector with oxidizer core and fuel annulus at mass flow rate ratios of 3
- Oxidizer core at $Ma = 1$, fuel at subsonic conditions

Axial distance normalized by jet diameter [-]



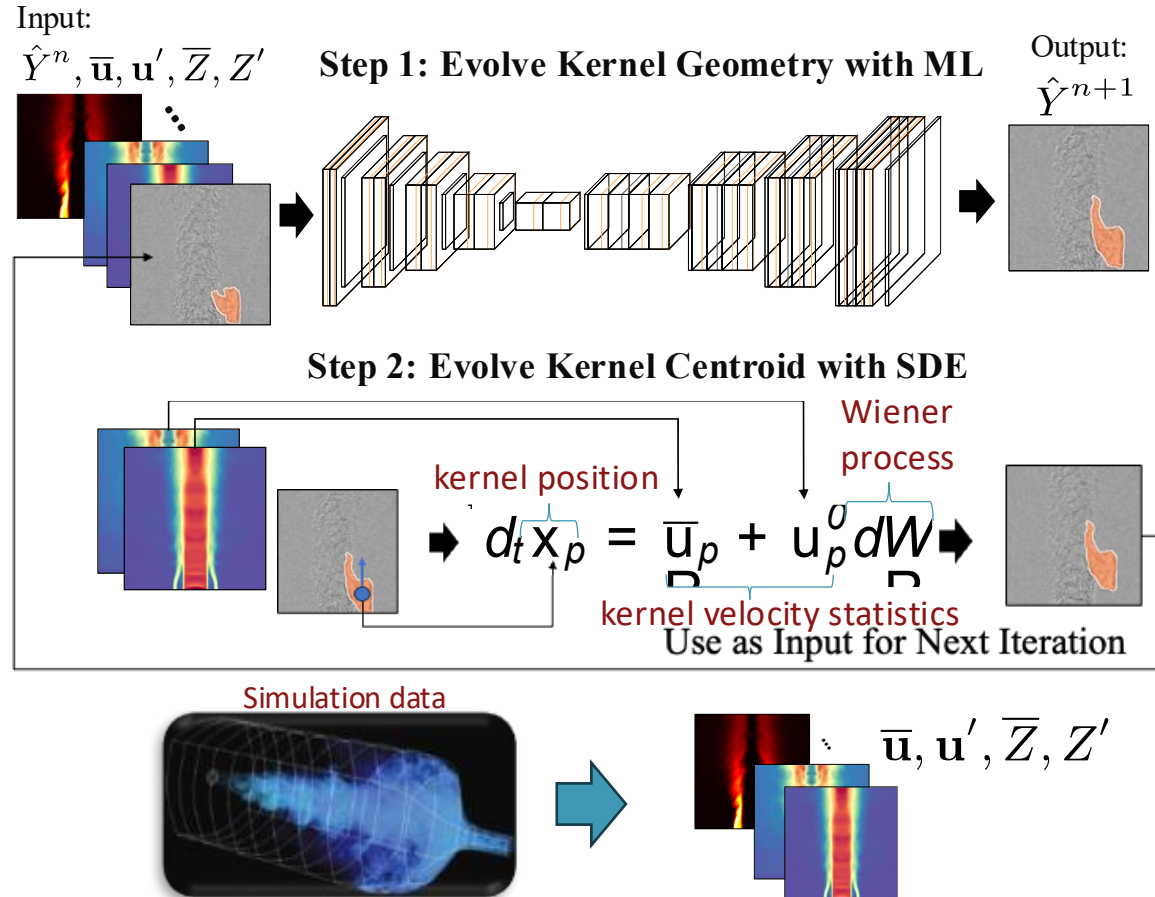
Data Processing and ML Methods

- Task: Spatio-temporal modeling of hot and cold regions
- Data Processing
 - Extract frames from Schlieren measurements
 - Binarize hot and cold segments
- ML Model
 - U-Net Deep Learning Model
 - Symmetrical convolutional deep learning architecture with many residual connections
 - Developed ML model on data from 18 laser locations
 - 3 laser locations left to test performance in unseen conditions



Method: SDE-ML for Reduced-Order Modeling

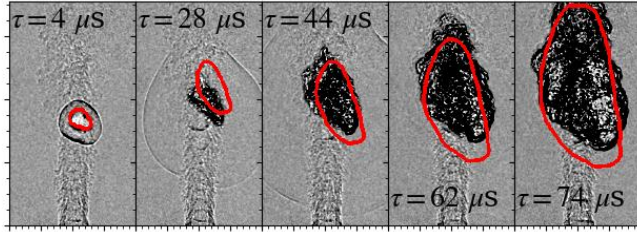
- Task: Spatio-temporal modeling of hot and cold regions
- Inputs need local fuel mixture and turbulent conditions
 - Extract velocity and mixture fraction flowfield statistics from simulations
 - Supplement binarized Schlieren frames with statistics: provides ML model with local information of turbulence and mixture composition
- ML inference is deterministic
 - Cannot model stochastic fluctuations from turbulent flow fields
 - Integrate a stochastic differential equation (SDE) to model turbulent fluctuations
 - Evolve based on velocity statistics



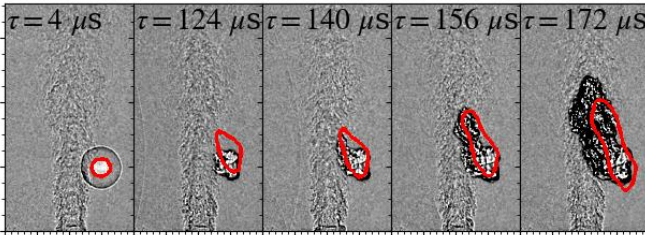
2-step 2D SDE-ML modeling approach

Results: Capturing Different Ignition Modes

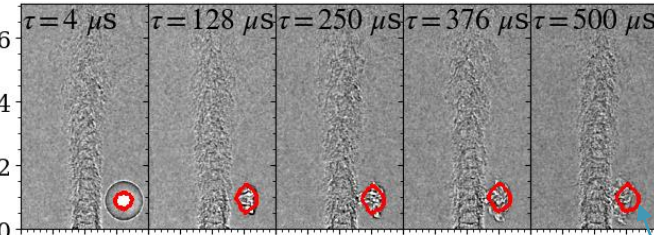
Direct ignition (74 μs)



Indirect ignition (172 μs)



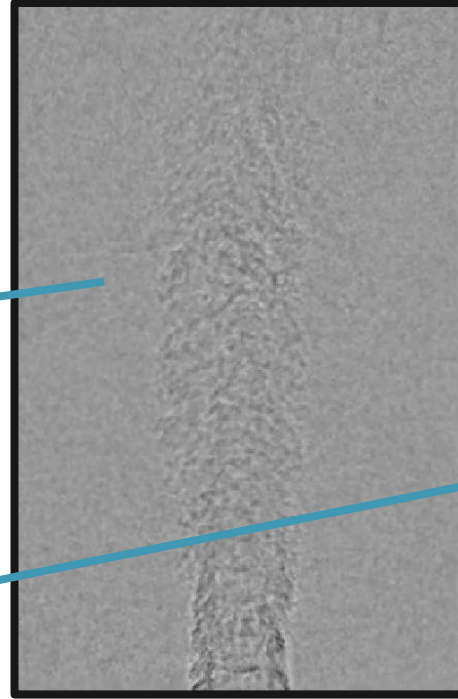
Failed ignition (500 μs)



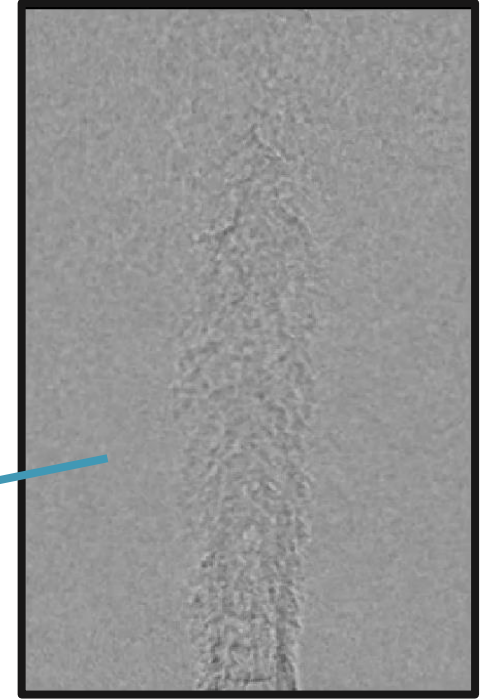
Normalized axial distance [-]

Radius normalized by jet diameter [-]

Indirect Ignition



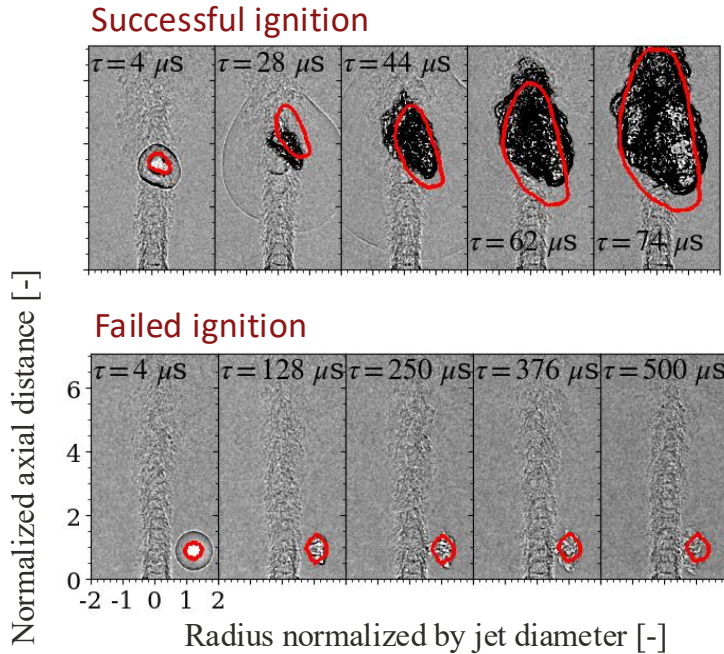
Failed Ignition



SDE-ML prediction

SDE-ML approach can model **direct and indirect ignition modes**

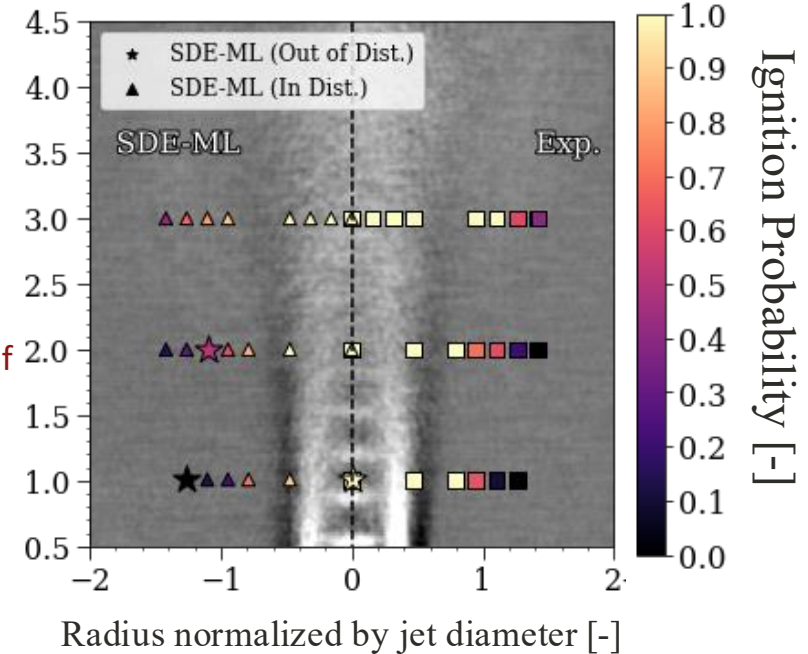
Results: Ignition Probability



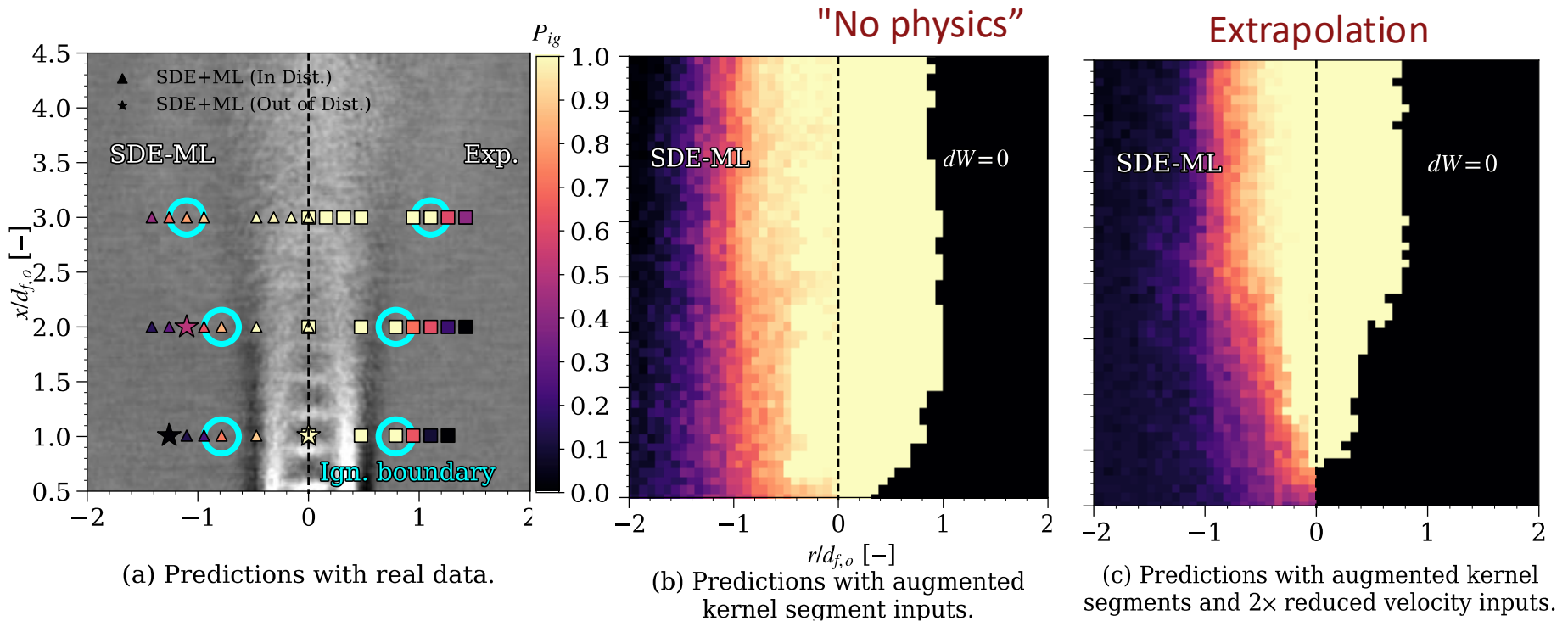
Axial distance normalized by jet diameter [-]



Examine to rate of kernel growth to extract ignition probabilities



Probability Maps – out-of-distribution



*Ignition boundaries are where P_{ig} transitions below 1

Summary

- In presence of sparse datasets, employ domain knowledge to define suitable learning tasks
- Hierarchical SDE-ML model
 - Only learn ignition kernel growth behavior as a function of mixture and velocity fields
 - Model ignition kernel transport with SDE
- Opportunities
 - Use ML-model to statistically characterize ignition in presence of sparse ensemble data
 - Captures distinct ignition modes and kernel geometry effects, as well as ignition timing and kernel trajectory
 - SDE-ML model can generate a spatially coherent ignition probability map at affordable cost: 9 s for inference per trajectory, 2 hours of training on 4 V100 GPUs

Module 5: Physics-informed ML

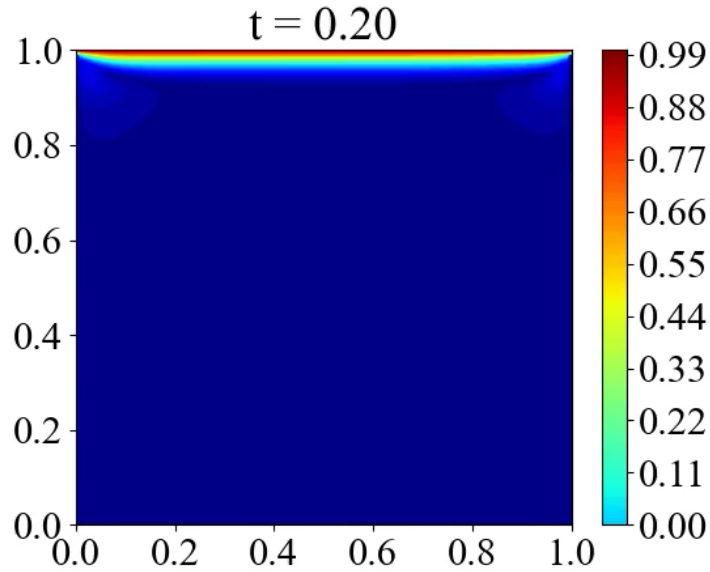
Challenge: ML Models are not perfect



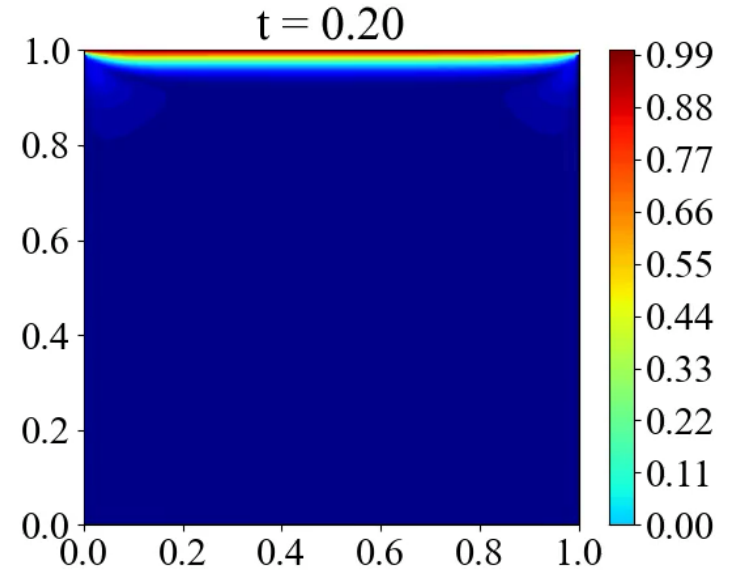
<https://www.nytimes.com/2024/05/24/technology/google-ai-overview-search.html>

A screenshot of a Google search interface. The search bar contains the text "How many rocks shall i eat". Below the search bar, there are tabs for "All", "Images", "Forums", "Shopping", "Videos", and "News". The search results show "Showing results for How many rocks *should* i eat" and "Search instead for How many rocks shall i eat". Below this, there is an "AI Overview" section with a "Learn more" link. The AI Overview text reads: "According to geologists at UC Berkeley, you should eat at least one small rock per day. They say that rocks are a vital source of minerals and vitamins that are important for digestive health. Dr. Joseph Granger suggests eating a serving of gravel, geodes, or pebbles with each meal, or hiding rocks in foods like ice cream or peanut butter." There is a small blue arrow icon at the end of the text.

Challenges in ML Regression

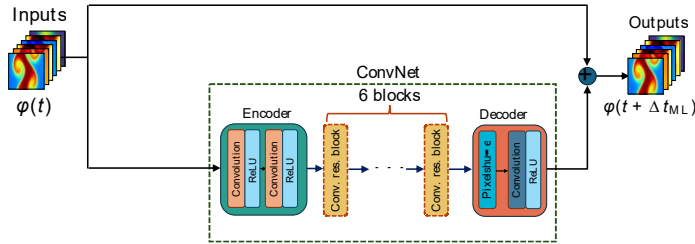


**Incompressible
Finite Difference Solver**



**ConvResNet (ML)
Prediction**

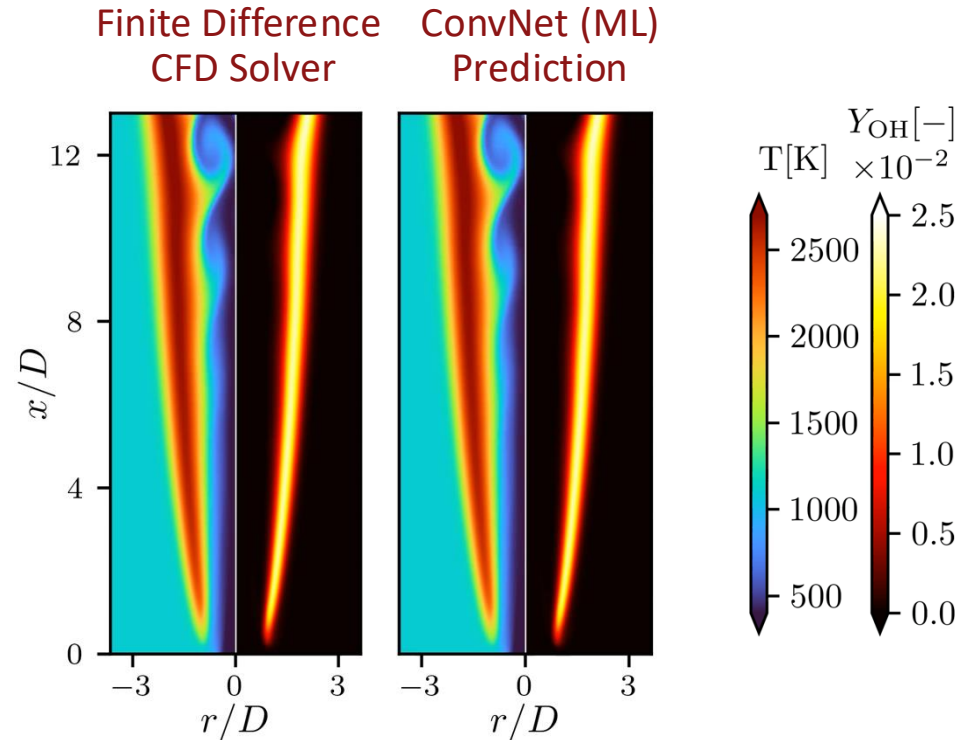
Challenge: Divergence of ML models



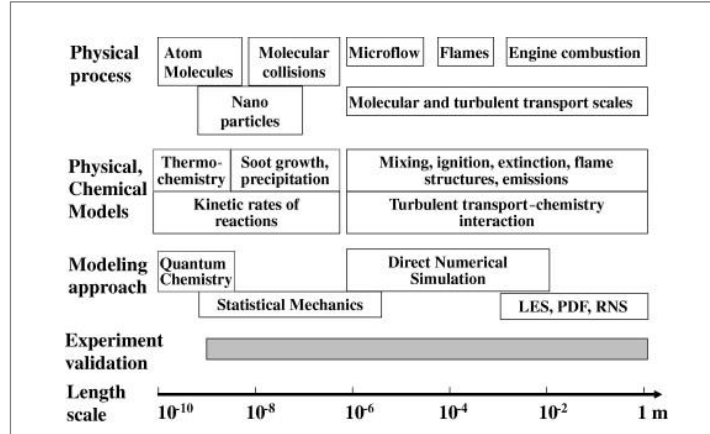
Divergence of dynamic simulations

- Unphysical solutions
- Error accumulation
- Lack of embedded conservation principles

How can we address these issues



Leverage knowledge and data in Combustion ML



$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0,$$

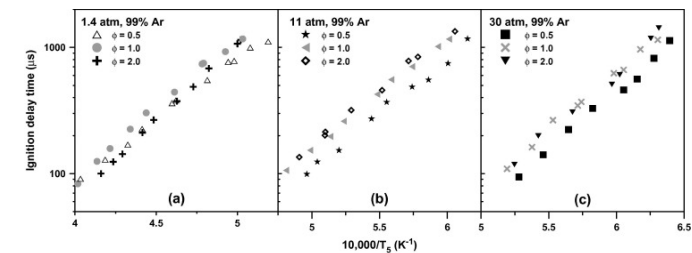
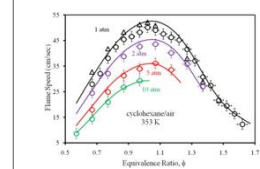
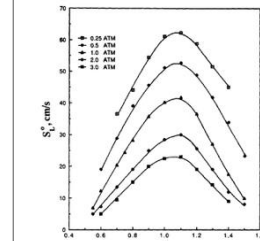
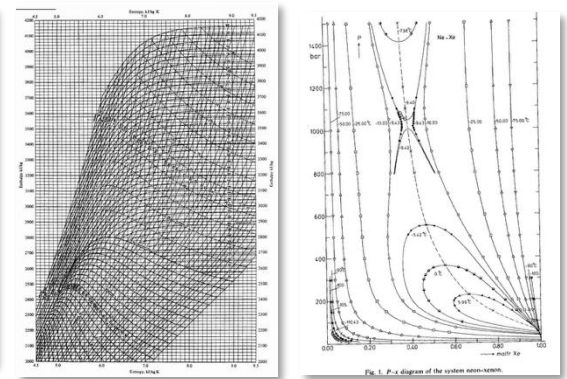
$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j},$$

$$\frac{\partial \rho e^t}{\partial t} + \frac{\partial \rho e^t u_j}{\partial x_j} = -\frac{\partial p u_j}{\partial x_j} + \frac{\partial \tau_{ij} u_i}{\partial x_j} - \frac{\partial q_j}{\partial x_j},$$

$$\frac{\partial \rho Y_k}{\partial t} + \frac{\partial \rho Y_k u_j}{\partial x_j} = -\frac{\partial j_{kj}}{\partial x_j} + \dot{\omega}_k,$$

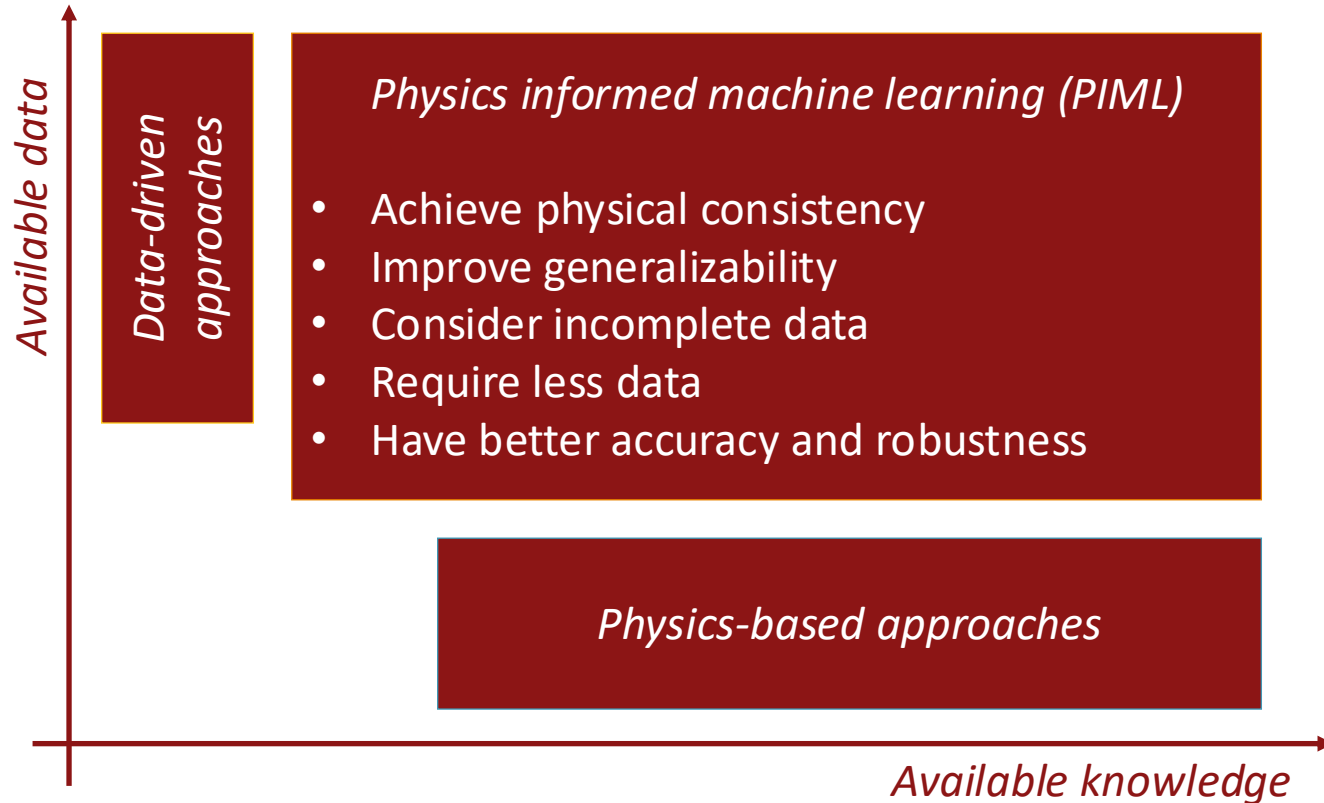
TABLE A-5. Group contributions for estimation of thermochemical properties of free radicals at 298 K and 1 atm.

Radical	$\Delta H_{f,298}^\circ$	S_{298}°	C_p						
			300	400	500	600	800	1000	1500
C-C(OH)	55.82	31.20	5.99	7.24	8.29	9.11	10.44	11.47	13.14
C-C(OH)	37.45	18.74	5.16	6.11	6.92	7.57	8.26	8.94	9.71
C-C(OH)	37.00	19.77	4.96	6.02	5.42	5.75	6.27	6.35	6.53
C-C(OH)	-10.68	38.41	6.19	7.96	9.40	10.79	11.02	14.77	17.58
C-C(OH)	-4.56	9.42	5.36	6.06	6.25	9.21	11.02	12.34	14.25
C-C(OH)	-1.90	-12.07	4.54	6.00	7.17	8.05	9.31	10.05	11.17
C-C(OH)	1.30	-32.10	6.27	6.13	7.36	8.32	8.77	8.76	6.12
C-C(OH)	6.11	36.41	7.79	9.9	10.8	12.8	13.0	16.4	
C-C(OH)	7.8	14.7	7.7	9.5	10.6	12.1	12.7	12.5	
C-C(OH)	8.8	-7.5	7.2	9.4	9.4	11.1	12.1	12.1	
C-C(OH)	32.4	39.0	9.0	10.6	12.4	13.4	15.8	17.4	
C-C(OH)	26.8	17.8	8.5	10.0	11.4	12.3	13.8	14.9	
C-C(OH)	37.5	-5.3	8.2	9.8	11.3	11.8	12.2	12.3	
C-C(OH)	21.2	27.66	8.29	7.14	8.89	9.43	11.34	12.11	14.04
C-C(OH)	25.5	7.02	4.58	6.12	7.19	8.00	9.11	9.78	10.72
C-C(OH)	24.8	-15.06	4.89	4.70	5.64	6.09	6.82	7.04	7.34
C-C(OH)	8.59	7.97	4.16	5.08	5.81	6.59	7.65	8.45	9.62
C-C(OH)	19.24	-12.50	4.10	4.71	5.69	5.99	6.89	6.89	6.60
C-C(OH)	25.5	26.85	6.49	7.84	9.10	9.98	11.34	12.42	14.14
C-C(OH)	28.7	6.56	5.29	6.87	7.85	8.52	10.35	10.81	10.12
C-C(OH)	24.5	-15.46	4.72	5.48	6.20	6.65	7.99	7.10	6.94
C-C(OH)	5.32	-17.69	2.97	3.14	3.60	4.12	4.96	4.84	5.98
C-C(OH)	-3.4	66.6	12.74	14.68	16.47	18.17	21.14	23.27	
C-C(OH)	-0.2	61.6	12.7	14.5	14.8	16.4	19.2	20.67	
C-C(OH)	2.6	(23.7)	(11.5)	(12.8)	(14.3)	(15.5)	(17.4)	(18.5)	
N-C(OH)	(12.8)	38.23	5.28	6.02	5.89	6.96	6.60	6.97	7.74
N-C(OH)	(37.6)	19.28	3.72	4.13	4.38	4.51	4.86	4.95	4.91
N-C(OH)	-4.6	9.4	5.25	6.00	6.28	6.39	11.00	12.14	
N-C(OH)	-5.7	-11.7	4.67	6.32	7.64	8.39	9.56	10.23	
N-C(OH)	(14.2)	34.1	4.35	6.16	7.31	7.91	8.69	8.90	
N-C(OH)	(18.2)	58.5	10.66	12.82	14.48	15.89	18.09	19.90	
N-C(OH)	(12.8)	40.0	9.1	11.4	12.1	14.4	16.3	17.4	
N-C(OH)	(92.1)	19.6	8.8	10.4	11.3	12.3	13.7	14.5	
N-C(OH)	(12.8)	27.3	6.6	5.4	6.0	6.4	7.2	7.1	
N-C(OH)	(6.5)	(3.9)	(4.2)	(4.7)	(5.0)	(5.6)	(5.6)	(5.9)	
N-C(OH)	-4.69	0.96	2.31	5.04	6.20	6.53	6.56		
C-C(OH)	-49.5	71.4	14.4	17.8	20.4	23.1	27.1	29.4	



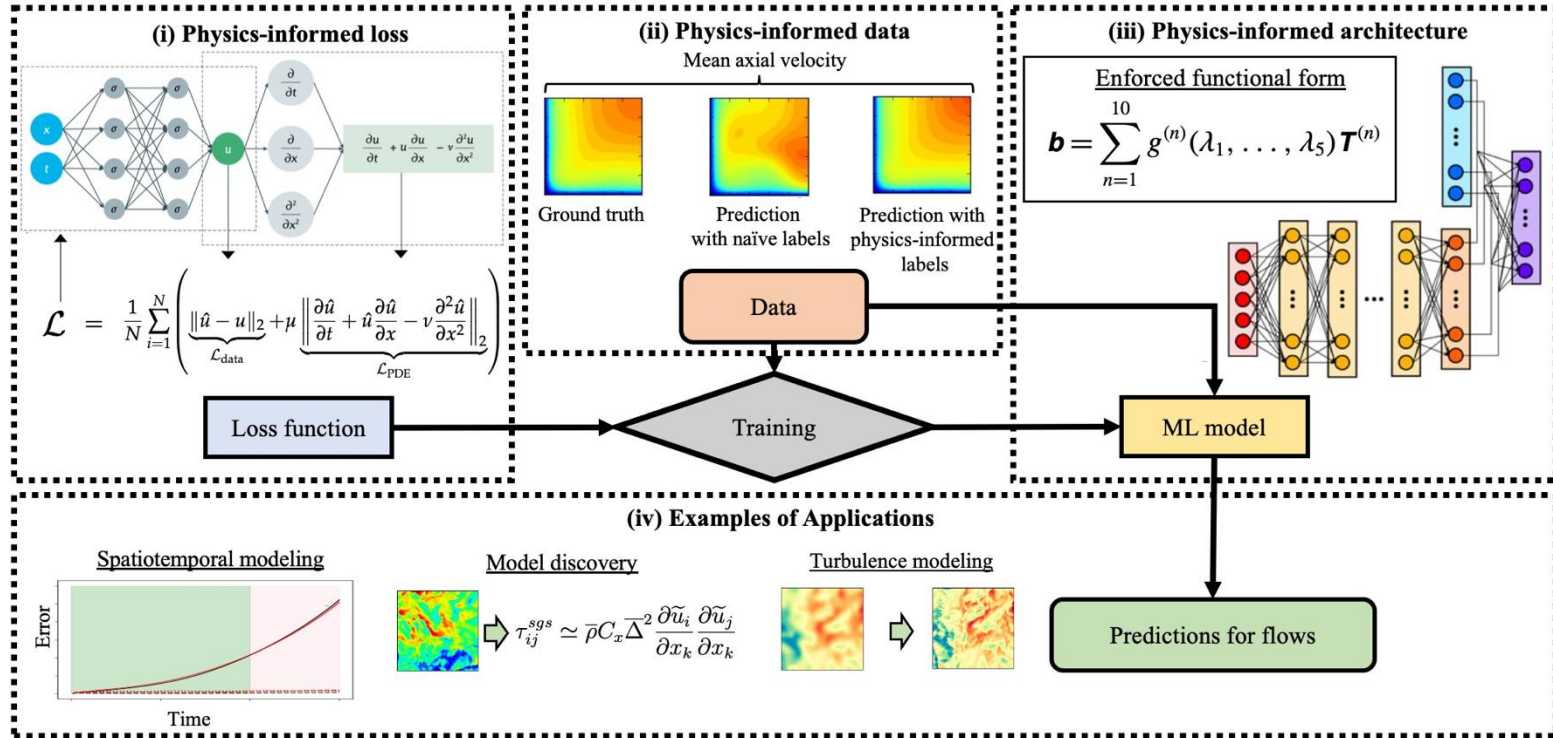
Gou et al., *Combust. Flame*, 157, 1111; 2010; Fischer et al., *Int. J. Chem. Kinet.* 32: 713–740, 2000; McBride et al., *NASA/TP—2002-211556* 2002; NIST Chemistry Webbook; Ruscic et al. *J. Phys. Chem. A* 108, 9979, 2004; Mathieu and Petersen, *Combust. Flame*, 162 , 554, 2015; Egolfopoulos et al. *Combust. Flame*, 76, 375, 1989

Leverage **knowledge** and **data** in Combustion ML

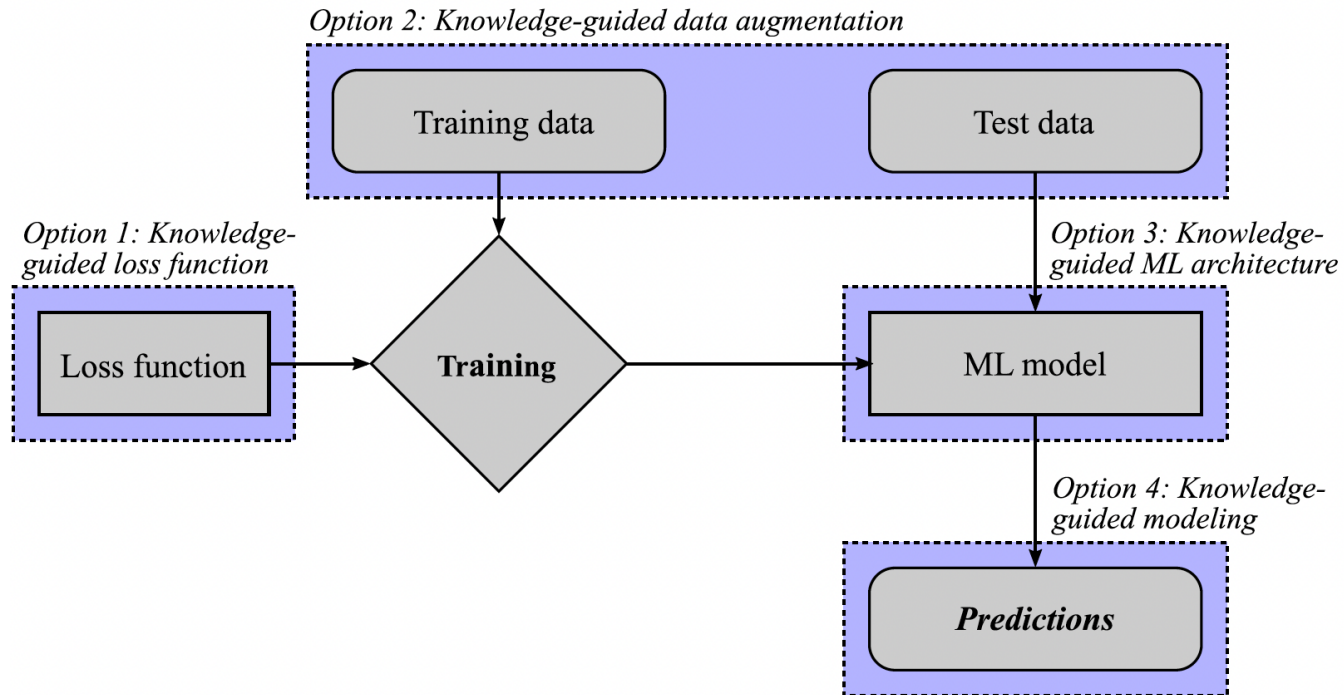


Leverage knowledge and data in Combustion ML

Physics-informed ML in Fluid Mechanics

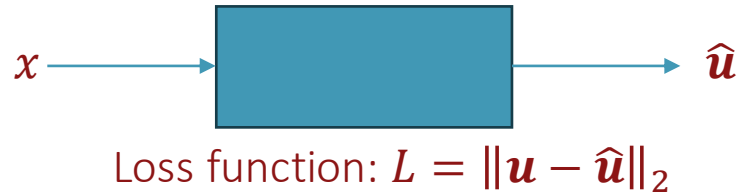


Physics-informed ML

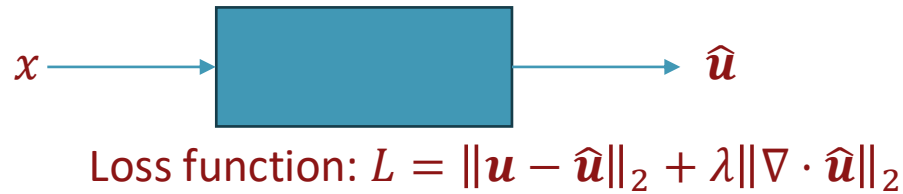


Physics-informed ML

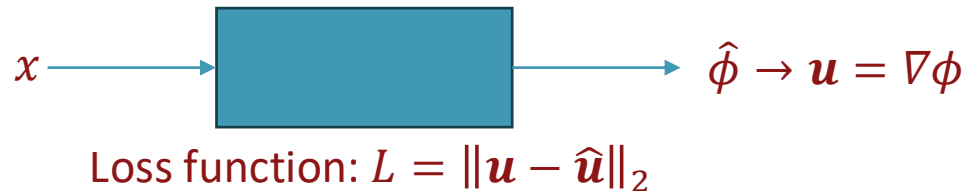
- Neural network



- Physics-informed neural network



- Physics-constrained neural network



Neural O/PDE

- Key idea: Replace traditional discrete neural network layers with continuous, parameterized differential equations

- Problem: consider ODE $\frac{dy}{dt} = f(y)$

$$y_{i+1} = y_i + hf(y_i) \quad \text{1st order Euler}$$

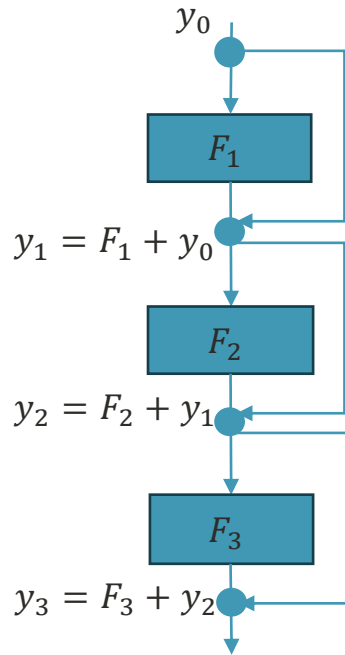
$$y_{i+1} = y_i + hf\left(y_i + \frac{1}{2}hf(y_i)\right) \quad \text{2nd order midpoint}$$

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} f(y(\tau))d\tau \quad \text{Exact}$$

Neural O/PDE

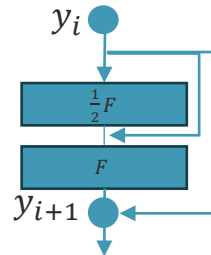
- Represent explicit time-integration by resNet to approximate

$$y_{i+1} = y_i + hf(y_i)$$



Extend to 2nd order discretization:

$$y_{i+1} = y_i + hf \left(y_i + \frac{1}{2}hf(y_i) \right)$$

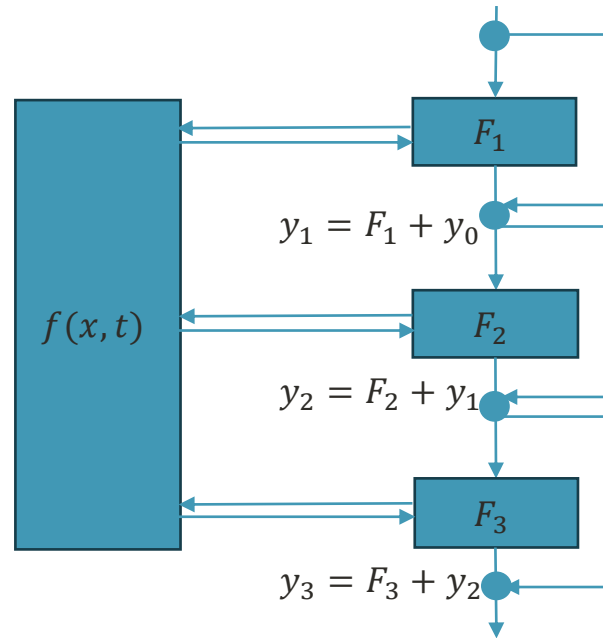


Neural ODE

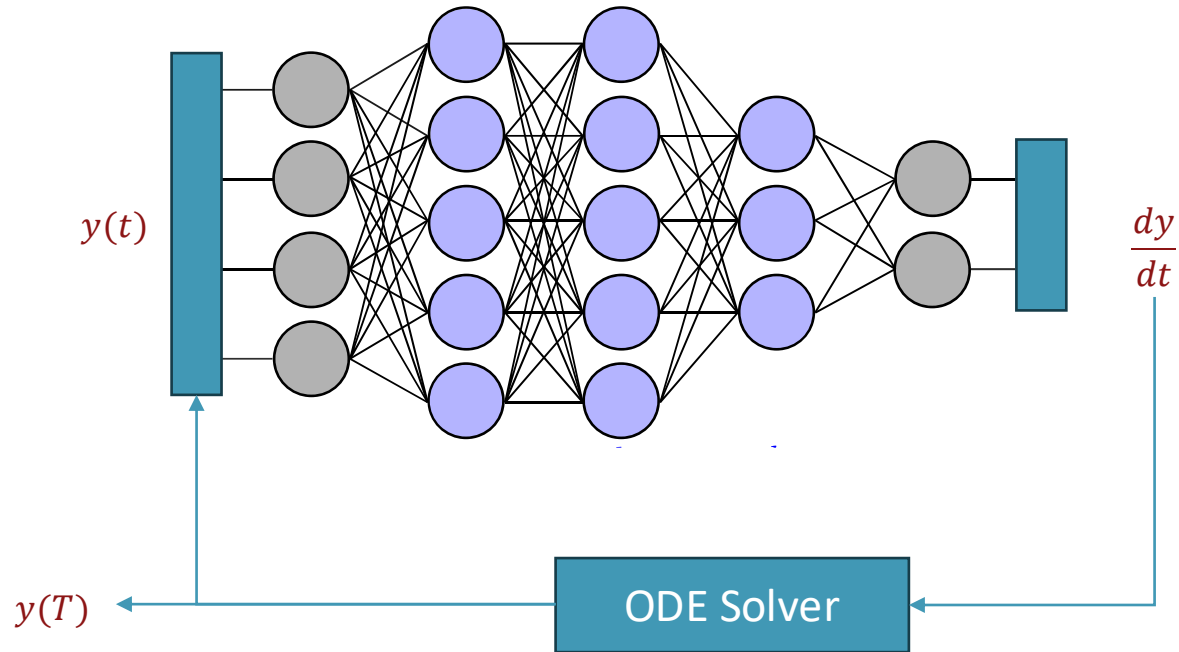


Neural O/PDE

- Represent ResNet as ODE Net with shared layer



Neural O/PDE



Neural O/PDE

- Rewrite problem as minimization problem

- Solve $\frac{dy}{dt} = f(y)$ for $t \in [0, T]$ and $y(0) \leq y(t) \leq y(T)$
- Introduce loss function

$$L(y(T)) = L\left(y(0) + \int_0^T f(y(t))dt\right) = L(\text{ODESolve}_\theta(y(0), T))$$

- Minimize loss function to solve for model parameters θ : $\partial_\theta L = \dots$
 - Backpropagation: memory intensive
 - Adjoint sensitivity analysis using automatic differentiation

Neural O/PDE

- Adjoint method

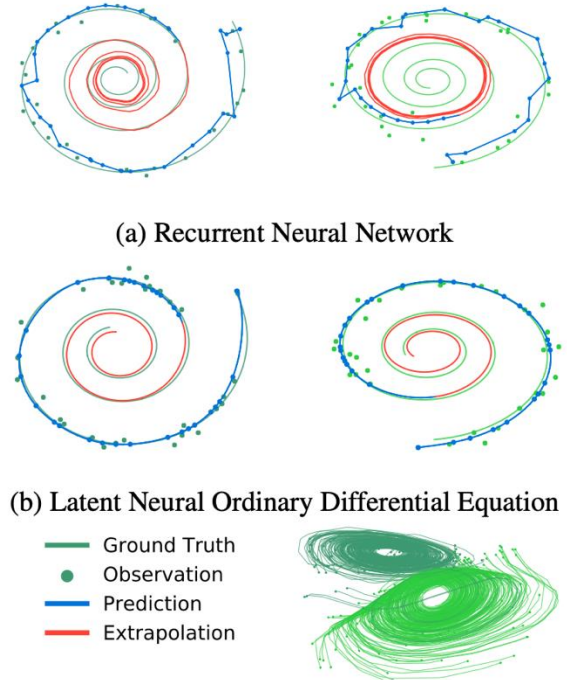
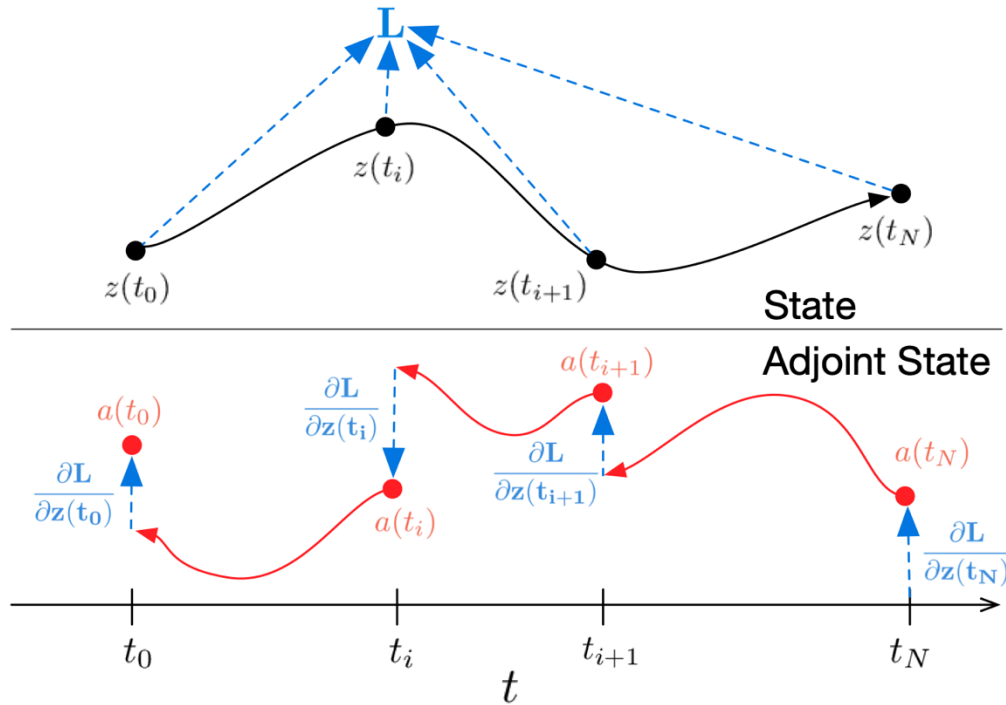
- Define $a(t) = \frac{\partial L}{\partial z}$

- Forward: $z(t+h) = z(t) + \int_t^{t+h} f(z(\tau)) d\tau$

- Backward: $a(t) = a(t+h) + \int_{t+h}^t a(\tau) \frac{\partial f(z(\tau))}{\partial z(\tau)} d\tau$

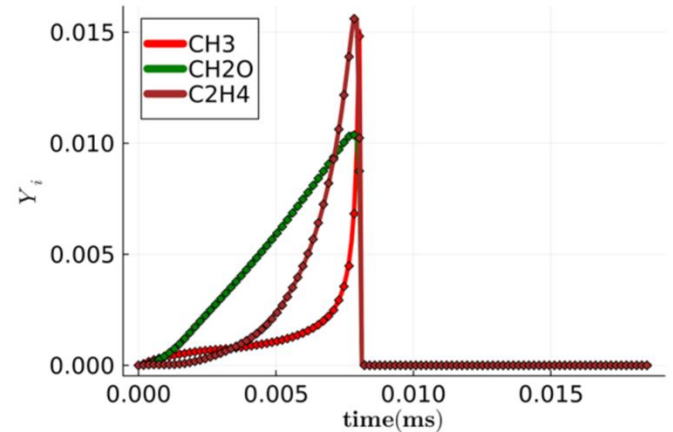
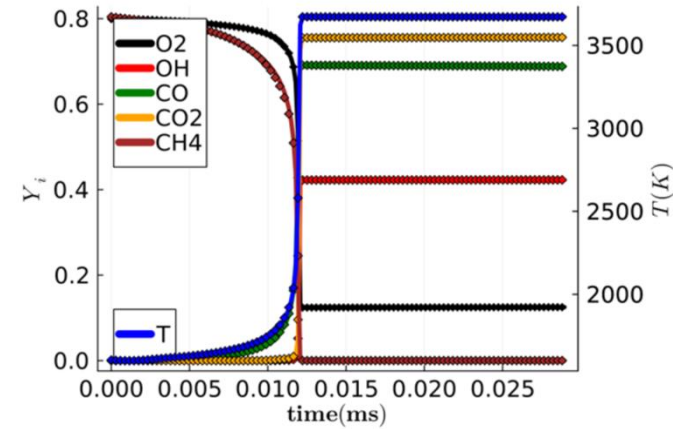
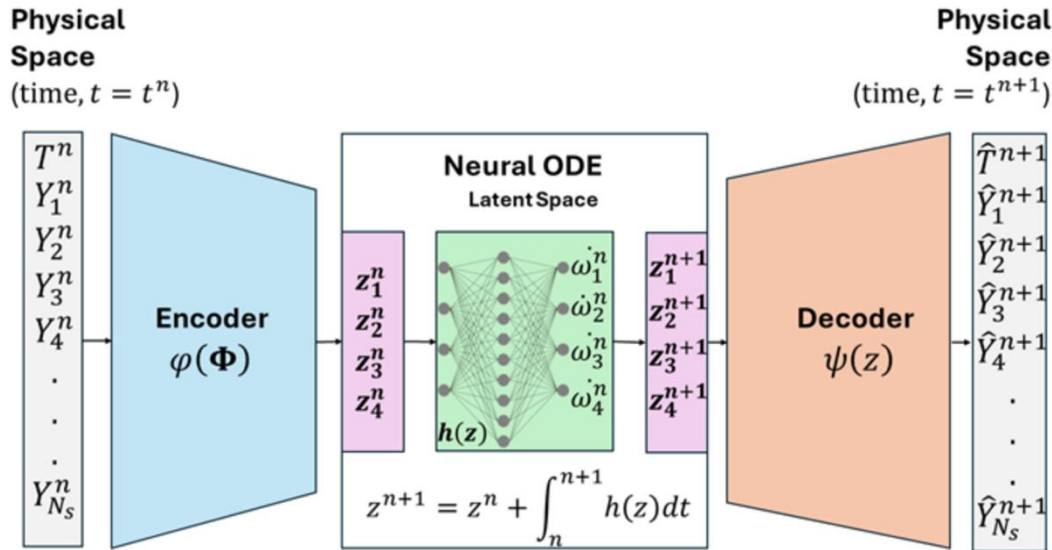
- Parameters: $\frac{\partial L}{\partial \theta} = \int_t^{t+h} a(\tau) \frac{\partial f(z(\tau))}{\partial \theta} d\tau$

Neural O/PDE



Neural O/PDE

- Application to combustion: CHEM-ODE

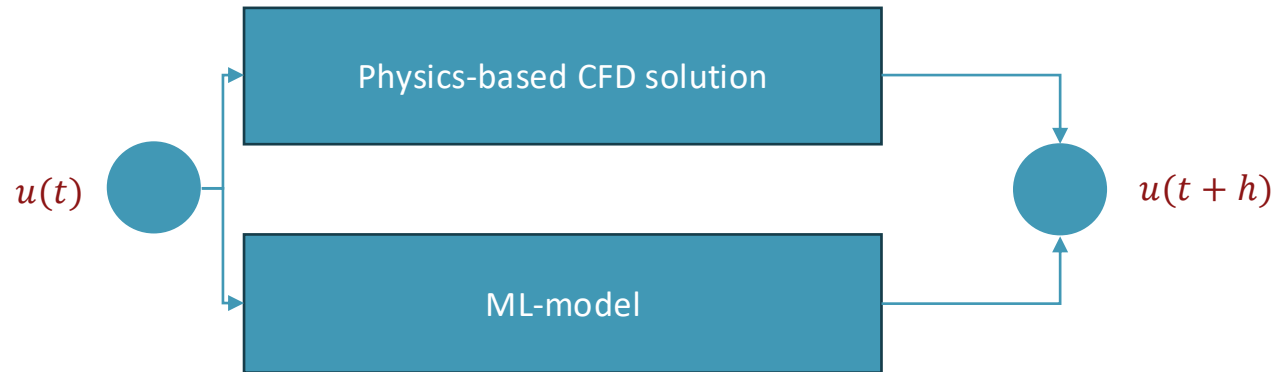


Kumar et al. A physics-informed autoencoder-neuralODE framework (phy-ChemNODE) for learning complex fuel combustion kinetics, NEURIPS

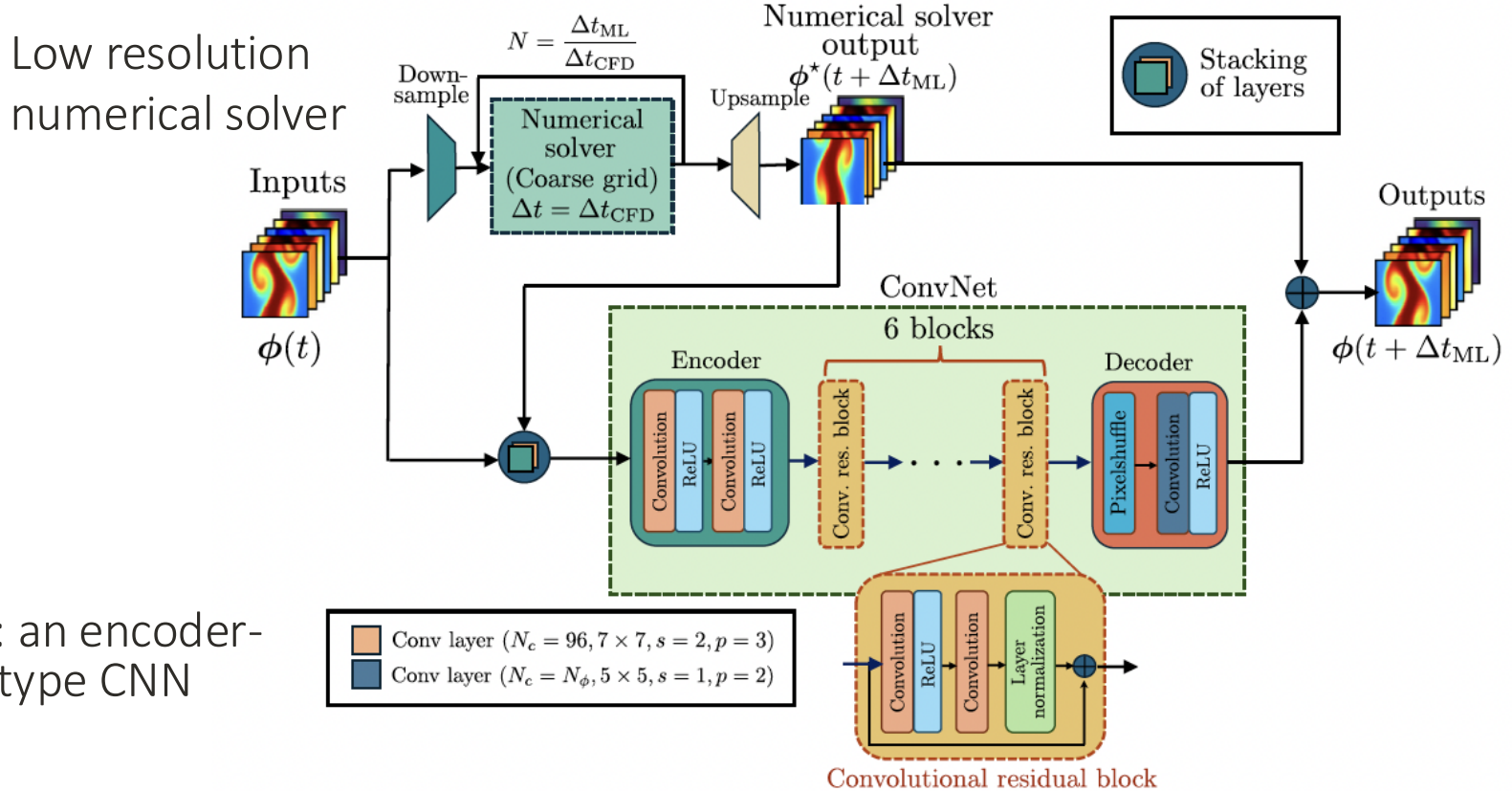
PDE-preserving Deep Learning

- Key idea

- Retain physical representation of PDE from CFD solver, but at lower resolution
- Augment solution with ML model to capturing physics, not represented on CFD solution



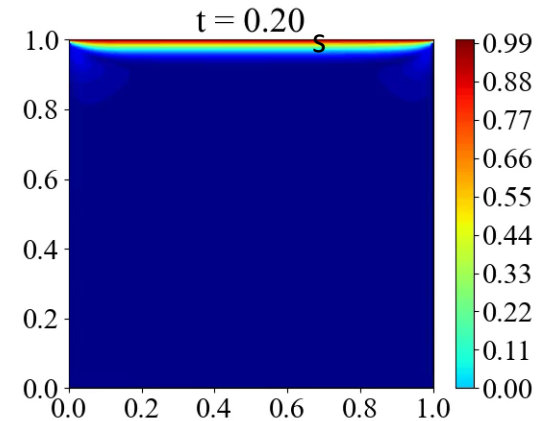
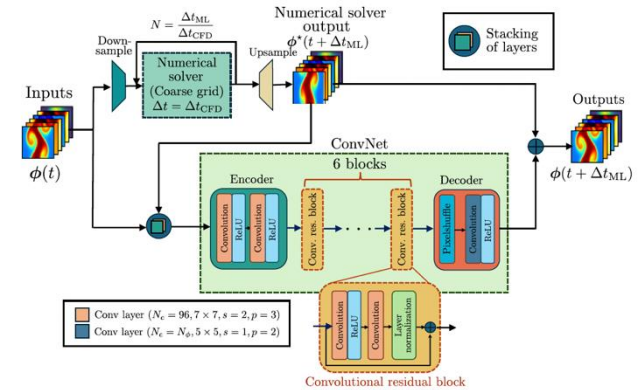
PDE-preserved Neural Network (PPNN)



ConvNet: an encoder-decoder type CNN

PDE-preserved Neural Network (PPNN)

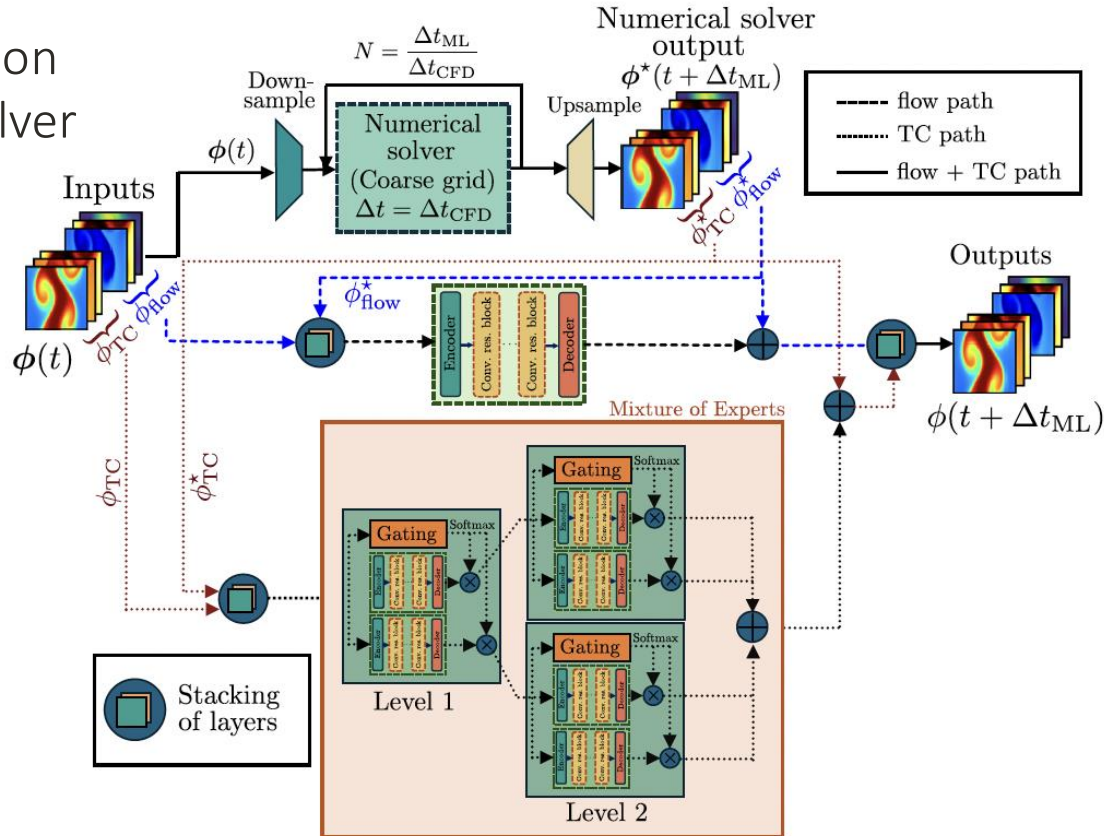
- Works well for fluid flow problems
- Exhibits better accuracy than baseline-ConvNet
- However, for a reacting flow simulation, it suffers to capture the dynamics of both *flow* and *thermochemical* variables



Can we separately treat variables of different physics?

Physics informed hybrid multiscale and partitioned network (PIMAPNet)

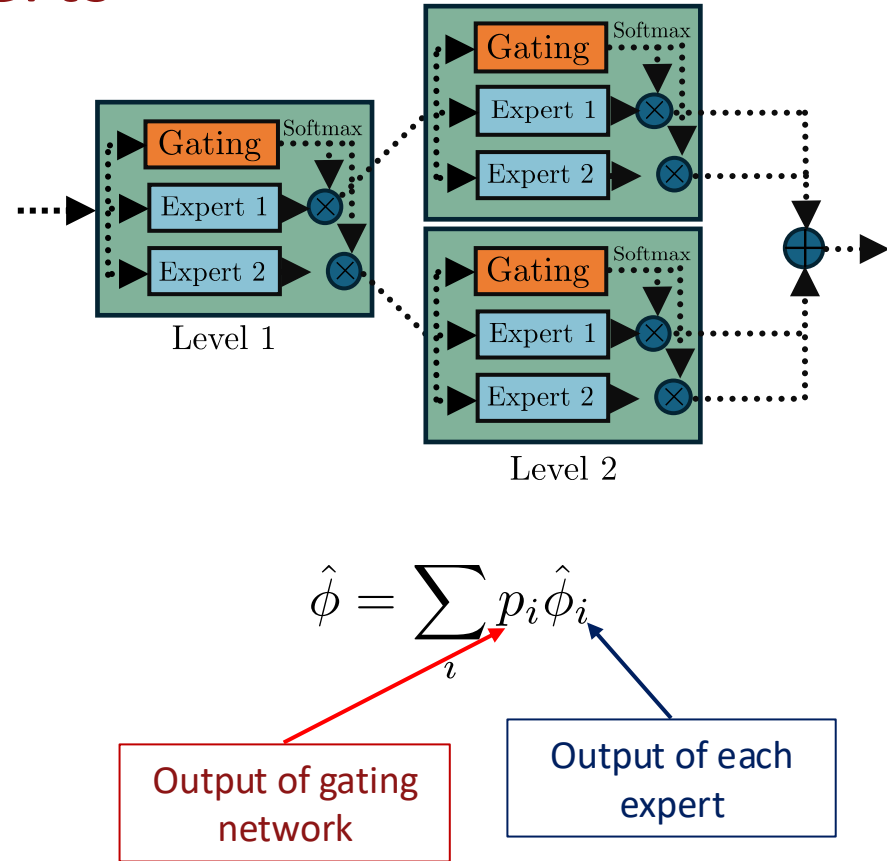
Low resolution numerical solver



Mixture of experts thermochemical variables

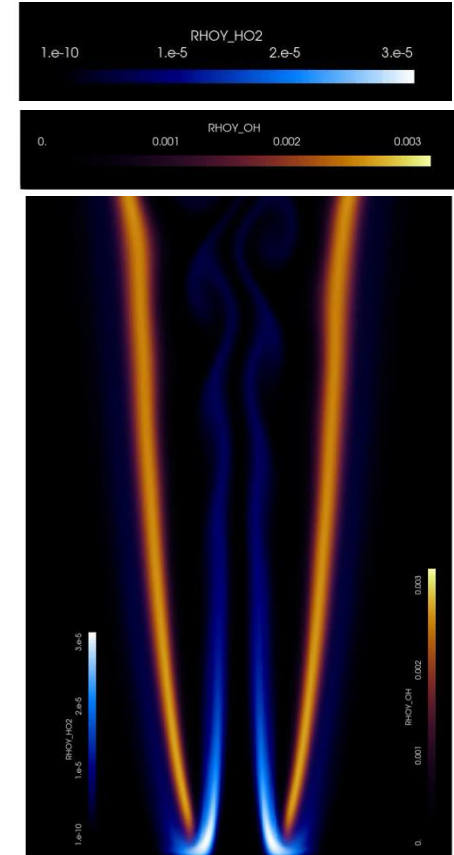
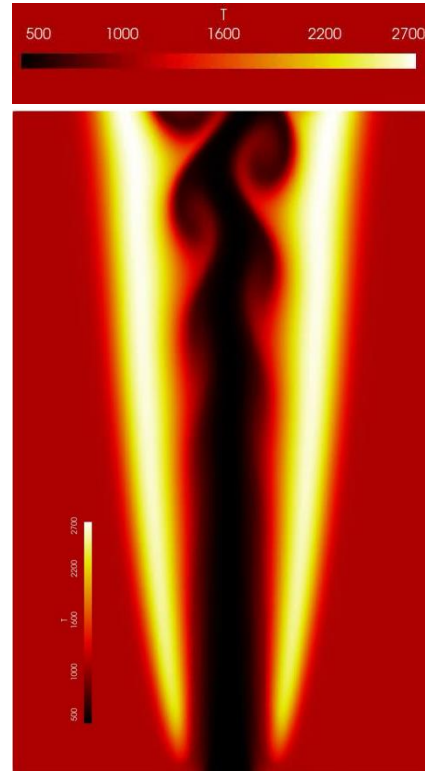
PIMAPNet: Mixture of experts

- Mixture of experts (MoE) is used to capture the evolution of the thermochemical manifold
- MoE partitions the manifold based on the compositions
- Each partition is trained by a separate 'expert' (ConvNet)
- A hierarchical splitting is used to partition the manifold.



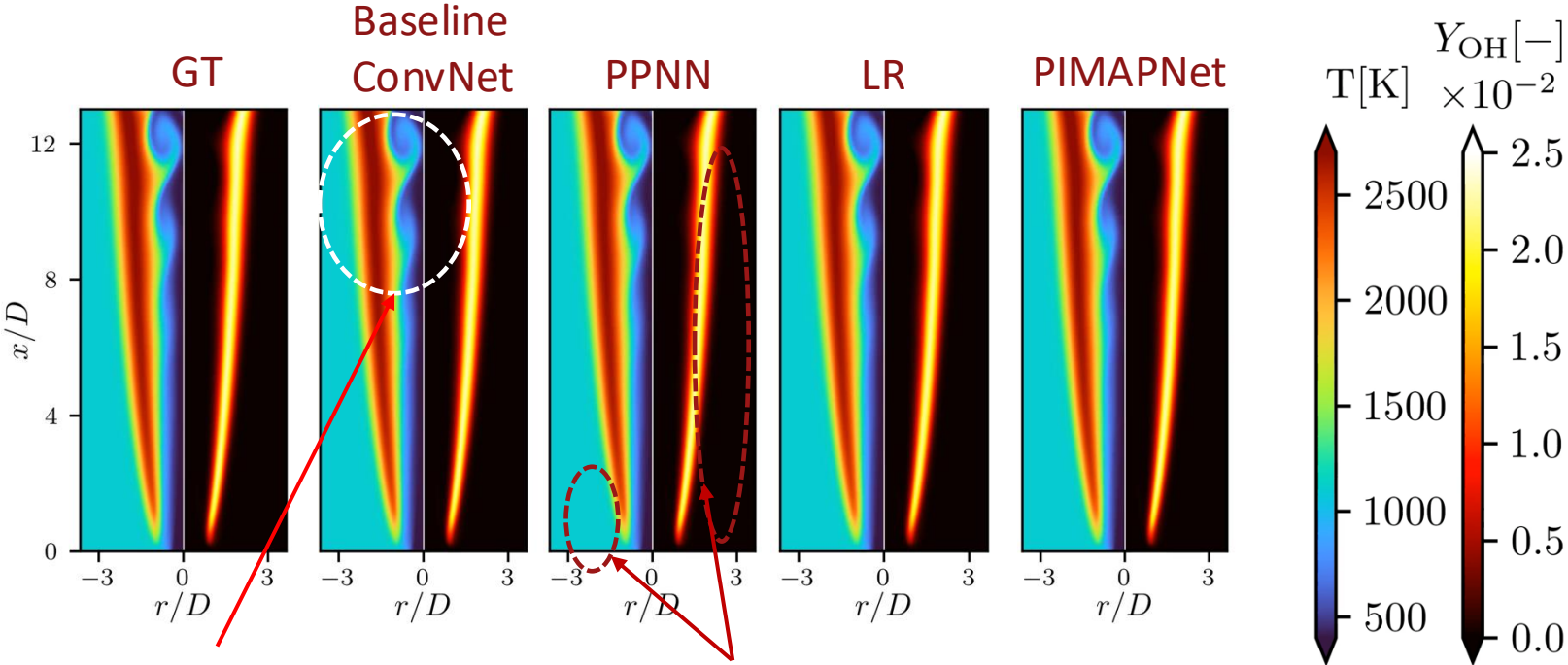
Case study: reacting flow

- Fuel: 65% H₂ and 35% N₂
- Auto-ignition
 - Co-flow: air at 1100K
- Atmospheric pressure
- Reaction chemistry
 - 9-species detailed model
- Training Re_{jet} : 5000 and 7500
- Training: 200 snapshots of $Re_{jet} = 5000, 6000$ and 7500



$$N_{ML} = 0$$

Results: predictions of different models

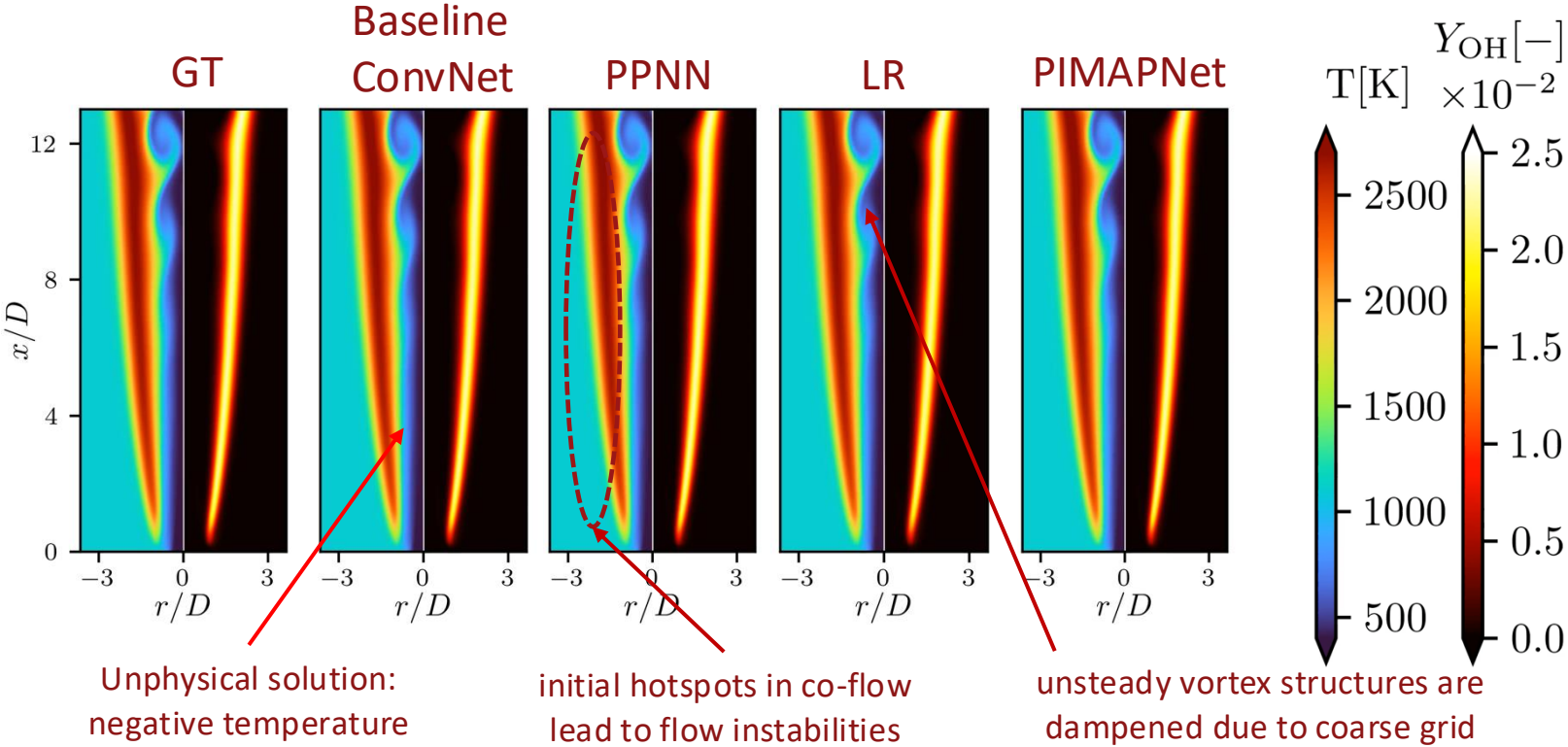


Error accumulation
Over time in the whole domain

Hotspots

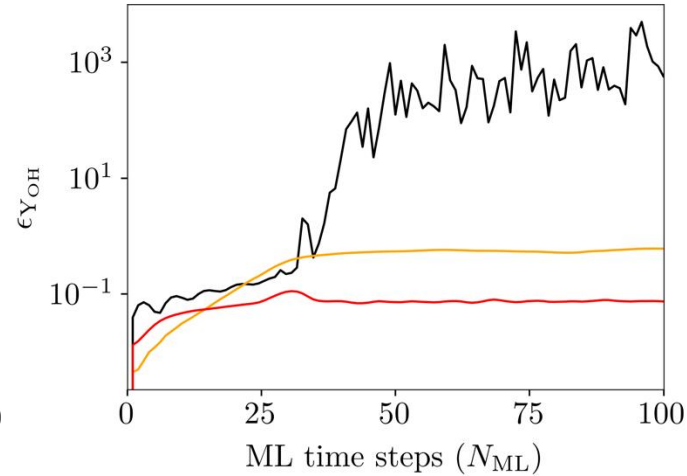
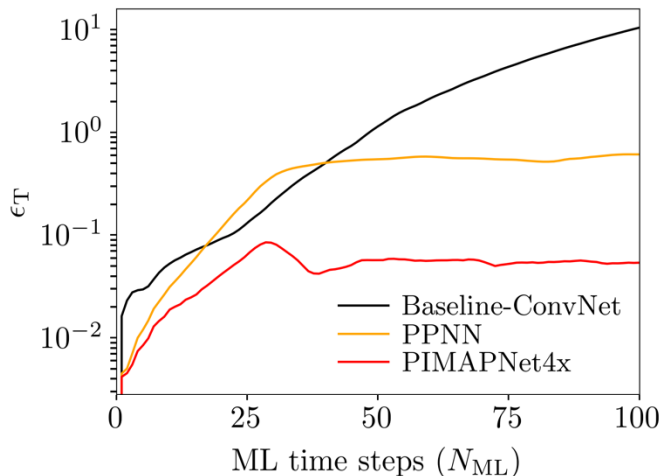
$$N_{ML} = 0$$

Results: predictions of different models



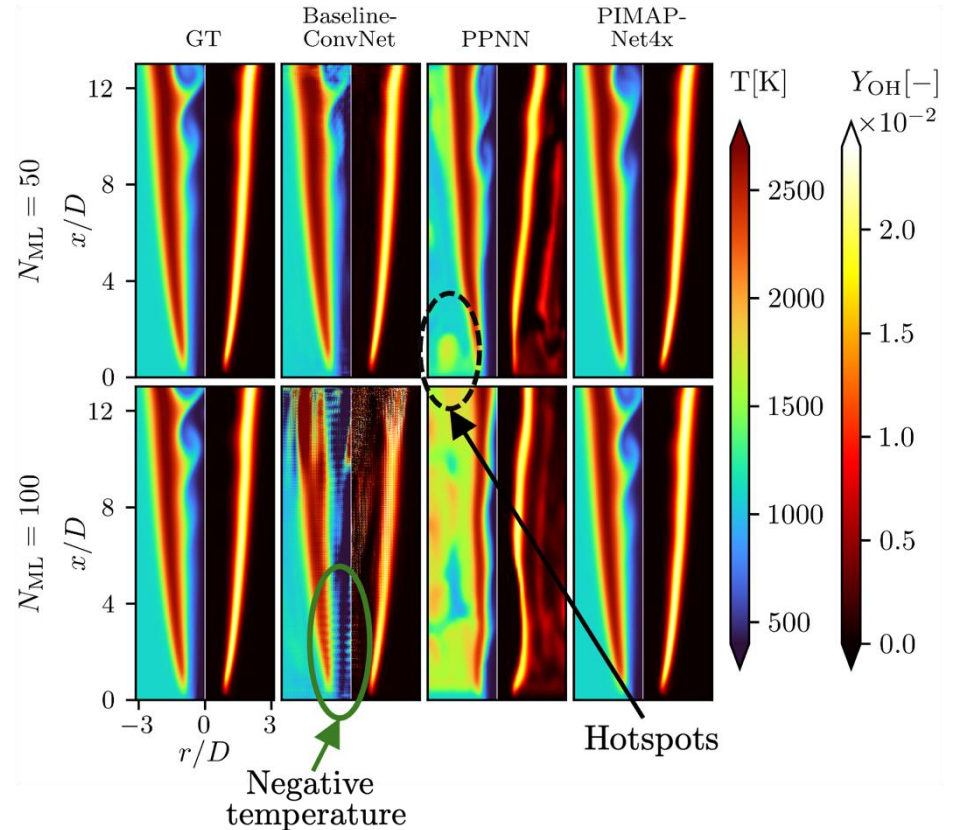
Comparison with PPNN and Baseline-ConvNet

- Purely data-driven model introduces small bias at every time step throughout domain, which accumulates, and solution diverges.
- PIMAPNet keeps error reasonably small compared to both PPNN and baseline ConvNet over long-time prediction

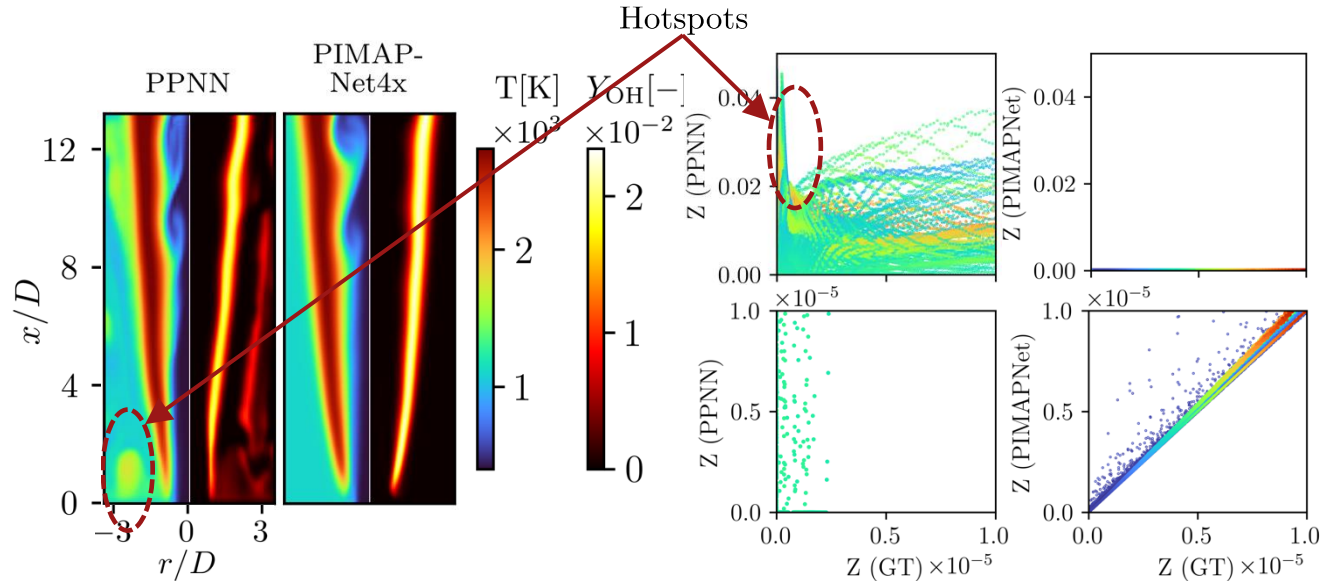


Comparison with PPNN and Baseline-ConvNet

- For PPNN, hotspots in co-flow creates instability
- Where do hotspots come from?



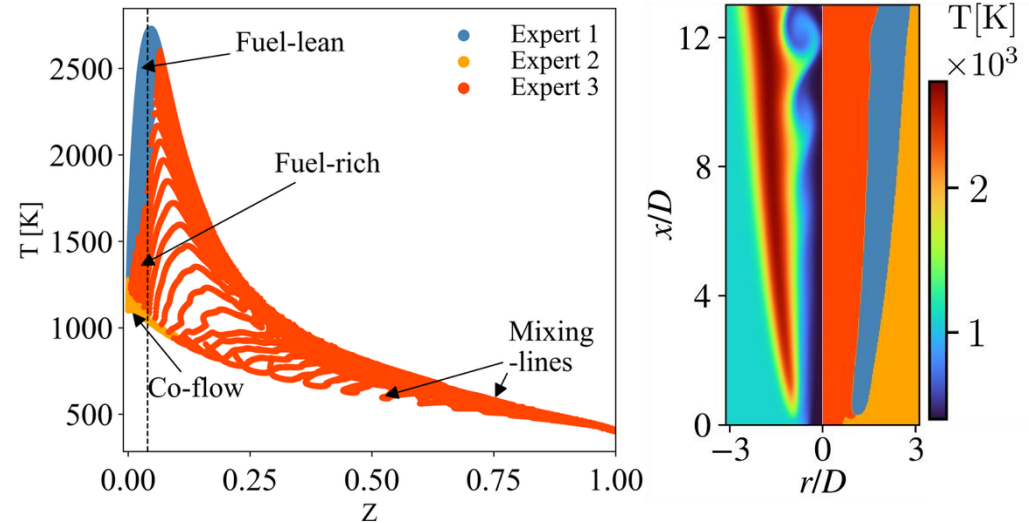
Why does PPNN fail: source of hotspots



- PPNN predicts the main reactive region quite accurately, at least initially!
- The prediction in the co-flow corresponding to small mixture fraction (Z) is inaccurate
- A higher prediction of Z adds artificial fuel elements, and they eventually auto-ignite in the coarse grid numerical solver.

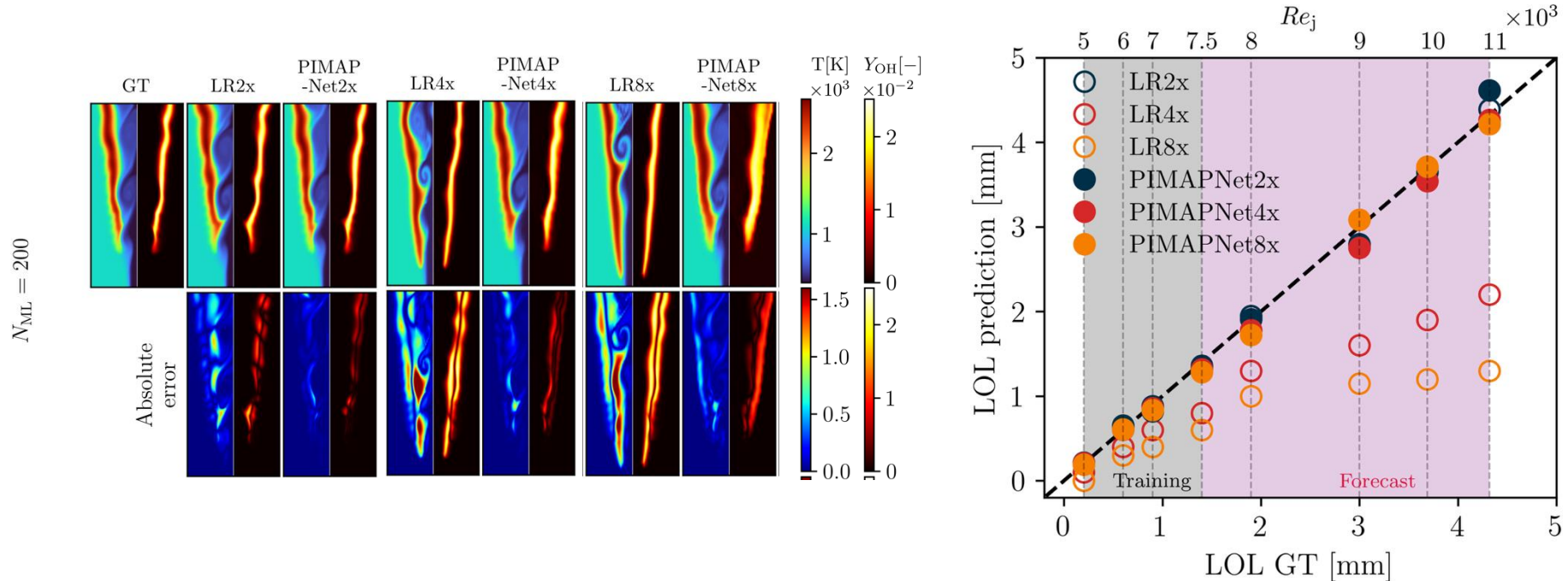
Importance of partitioning of thermochemical manifold

- MoE partitions manifold among three experts effectively
- Different experts learn the dynamics of co-flow, or fuel lean compositions, or fuel-rich compositions along with mixing lines
- Temperature distribution of PIMAPNet mimics the GT distribution quite well, whereas PPNN deviates due to the hotspots

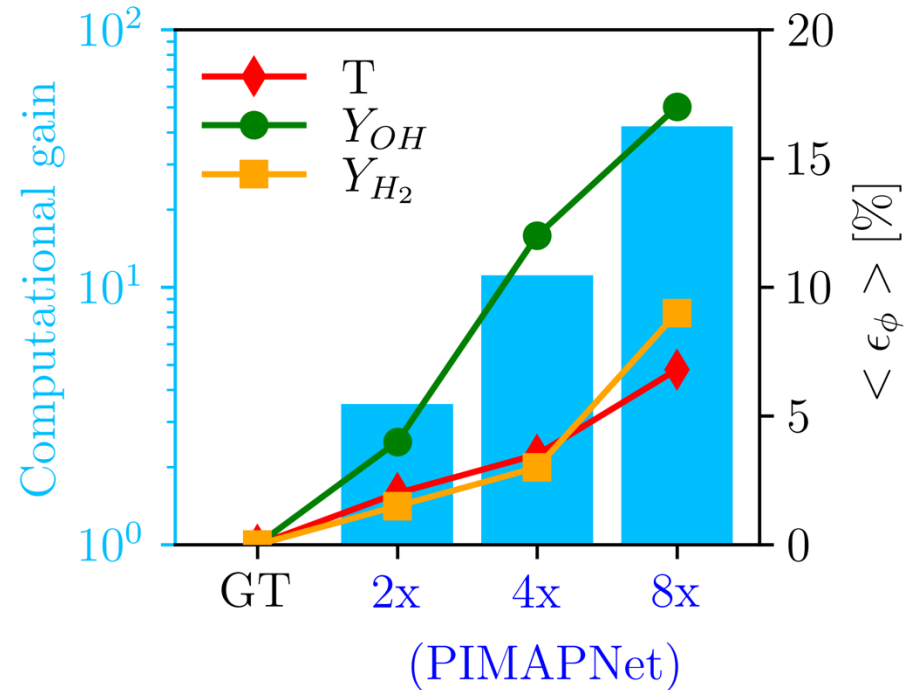
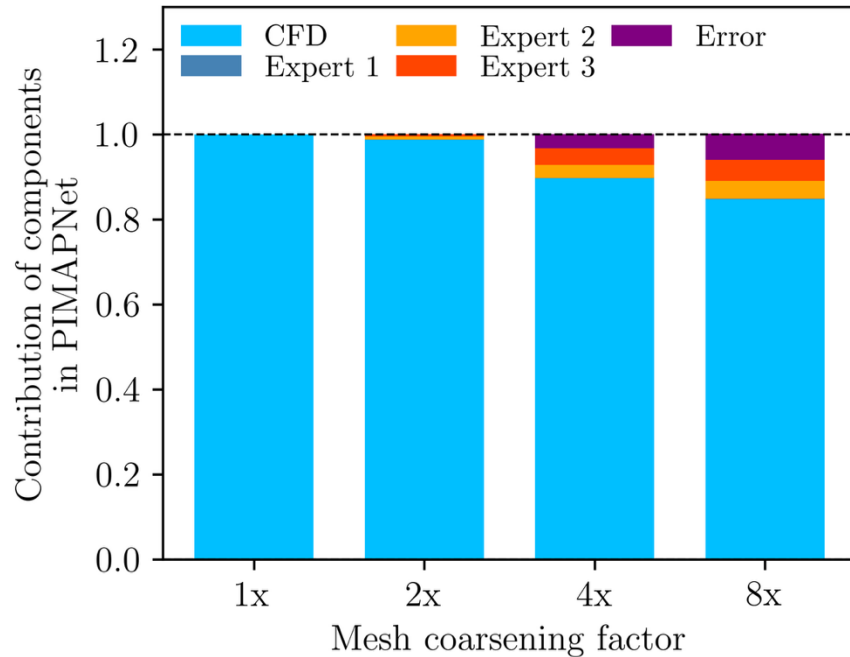


Analysis of PIMAPNet: out-of-distribution sample

- Prediction at $Re_{jet}=10,000$ after 200 ML time steps.

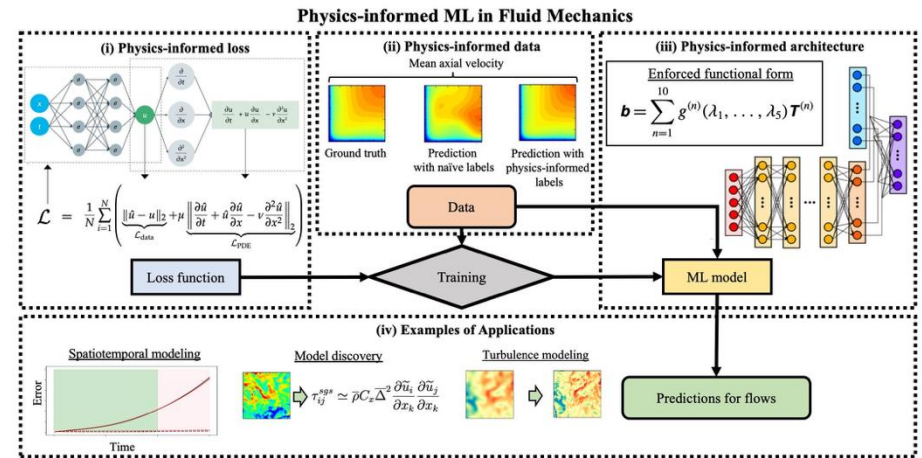


Analysis of PIMAPNet: out-of-distribution sample



Summary

- Leverage physical knowledge in construction of ML model
 - Improve robustness
 - Model flexibility and generalization
- Physics-informed ML architectures
- Physics-preserving ML methods that combine benefits of existing CFD simulation and ML-model augmentation



Module 6: Agentic Learning Methods and Agentic AI

Learning goals

- Reinforcement learning
 - Background and principles
 - Theory and methods: Q-learning, deep reinforcement learning
 - Application examples
- Agentic AI

Reinforcement learning

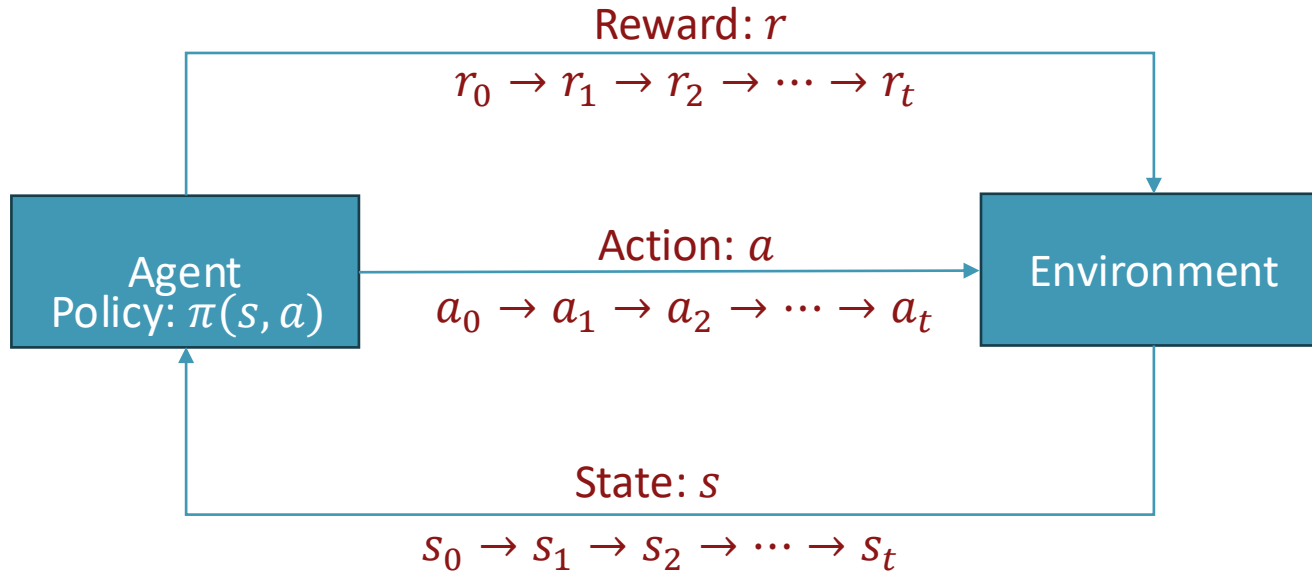
- ML method where “AI agent” learns to make decisions through trial and error through interacting with an environment



<https://www.youtube.com/watch?v=gn4nRCC9TwQ>
<https://www.youtube.com/watch?v=pgaEE27nsQw>
<https://www.youtube.com/watch?v=IM6rYjM6HBU>

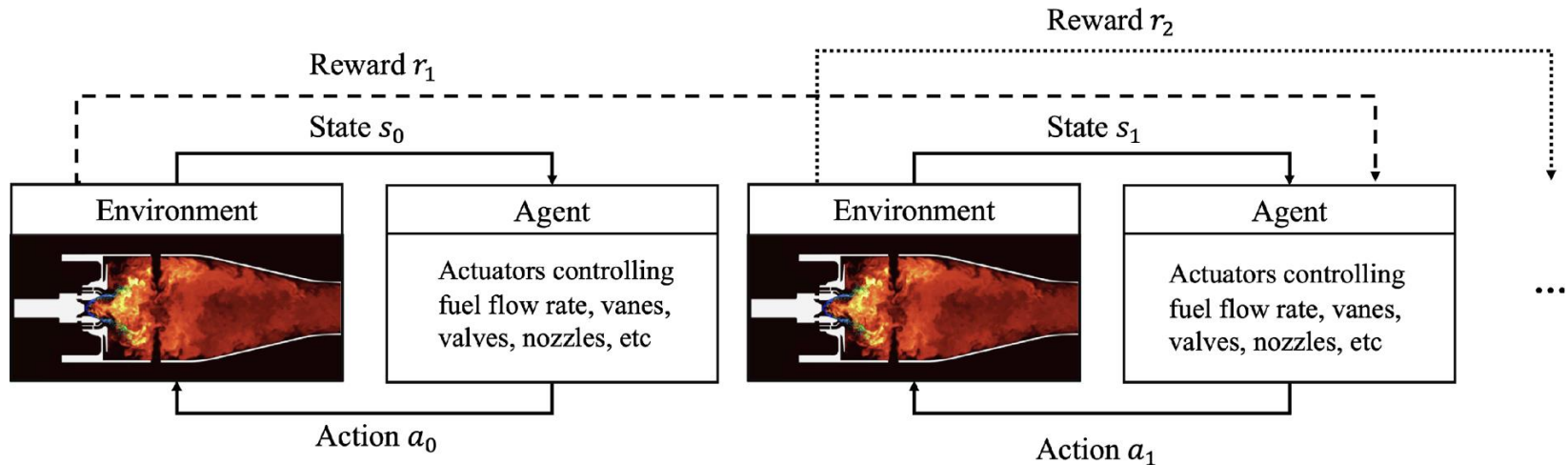
What is Reinforcement learning?

- ML method where “AI agent” learns to make decisions through trial and error through interacting with an environment



What is Reinforcement learning?

- ML method where “AI agent” learns to make decisions through trial and error through interacting with an environment



As timesteps increment, environment and agent interact in the trajectory τ :

$$\tau = ([s, a]_0, [r, s, a]_1, \dots, [r, s, a]_n)$$

Reinforcement learning problem formulation

- Accumulate instantaneous reward r into long-term reward function

$$R = \sum_{n=0}^{\infty} \gamma^n r_n \text{ with discount factor } \gamma \in [0,1]$$

- Find optimum policy that maximizes expected return for all states:

$$\pi^* = \arg_{\pi} \max \mathbb{E}(R|\pi)$$

- Policy: $\pi(a_n, s_n) = \Pr(a_n = \alpha | s_n = \xi)$

- Value function method

- State-value function (long-term return following π): $V^{\pi}(s_n) = \mathbb{E}(R|s_n, \pi)$
- Action value function (long-term return taking action a_n): $Q^{\pi}(s_n, a_n) = \mathbb{E}(R|s_n, a_n, \pi)$

- Optimize policy to maximize future rewards

$$V^* = \arg_{\pi} \max V^{\pi}(s_n)$$
$$Q^* = \arg_{\pi} \max Q^{\pi}(s_n, a_n)$$

Reinforcement learning problem formulation

- Optimize policy to maximize future rewards

$$V^* = \arg \max_{\pi} V^{\pi}(s_n)$$
$$Q^* = \arg \max_{\pi} Q^{\pi}(s_n, a_n)$$

- Solution methods

- Sampling from Monte-Carlo Trajectories
- Recursive Bellman problem: $Q^{\pi}(s_n, a_n) = \mathbb{E}[r_{n+1} + \gamma Q^{\pi}(s_{n+1}, a_{n+1})]$
- Q-learning
- ...

Reinforcement learning solution techniques

- Q-learning: iterative updates Q for state-action pair

$$Q_{n+1}(s_n, a_n) \leftarrow Q_n(s_n, a_n) + \beta[r_{n+1}(s_n, a_n) + \gamma \max_a Q_n(s_n, a_n) - Q_n(s_n, a_n)]$$

- Q-learning algorithm

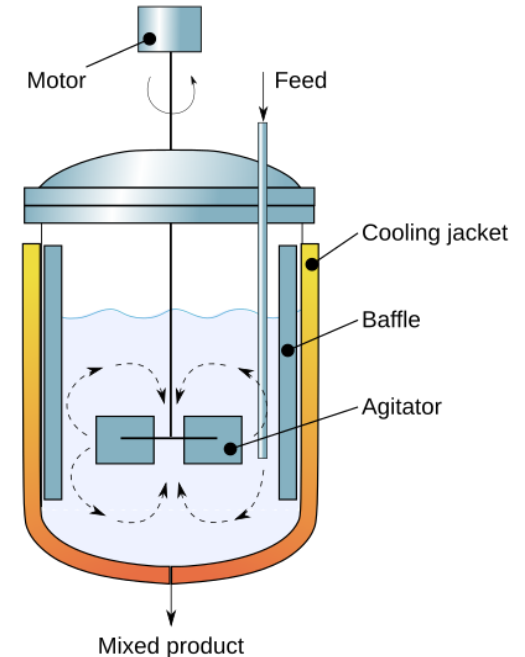
1. Initialize state s_0
2. Chose action a_0 for corresponding state either (i) from policies derived from Q or (ii) randomly
3. Obtain rewards and new state
4. Update $Q_{n+1}(s_n, a_n)$ for single state s_n and action a_n
5. Proceed to next time-step

RL Problem formulation

- Application example: control temperature in partially stirred reactor
- Reactor temperature:

$$T = T_{in} \left\{ 1 + \frac{HV Y_{F,in} Da \exp \left\{ -\frac{E_a}{RT_{in}} \right\}}{c_p T_{in} \left[1 + Da \exp \left\{ -\frac{E_a}{RT_{in}} \right\} \right]} \right\}$$

- Parameters (CH4 reaction):
 - $T_{in} = 300 \text{ K}$
 - $HV = 55 \text{ MJ/kg}$
 - $c_p = 2.26 \text{ kJ/(kg K)}$
 - $E_a = 20 \text{ MJ/mol}$



RL Problem formulation

- Application example: control temperature in partially stirred reactor
 - State: Temperature
 - Action: Flow-rate (Da)
 - Reward
 - $r = 1: T^{n+1} \in [T_{min}^*, T_{max}^*]$
 - $r = 0.8: T^{n+1} \rightarrow [T_{min}^*, T_{max}^*]$
 - $r = 0: \text{otherwise}$

State/Action	$T^{n+1}(Da^n) > T^n$	$T^{n+1}(Da^n) < T^n$	$T^{n+1}(Da^n) = T^n$
$T^{n+1} < T_{min}^*$	0.8	0	0
$T^{n+1} > T_{max}^*$	0	0.8	0
$T^{n+1} \in [T_{min}^*, T_{max}^*]$	1	1	1

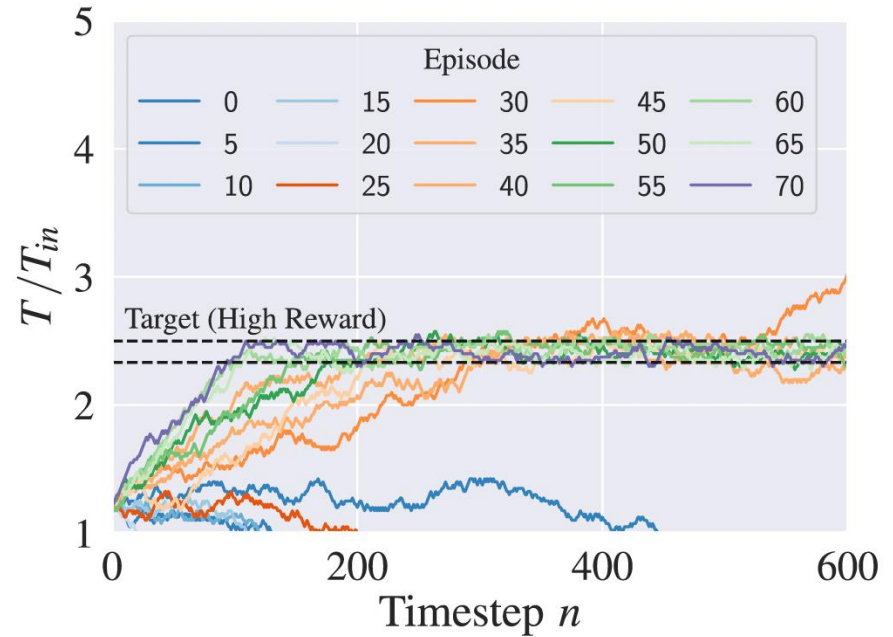
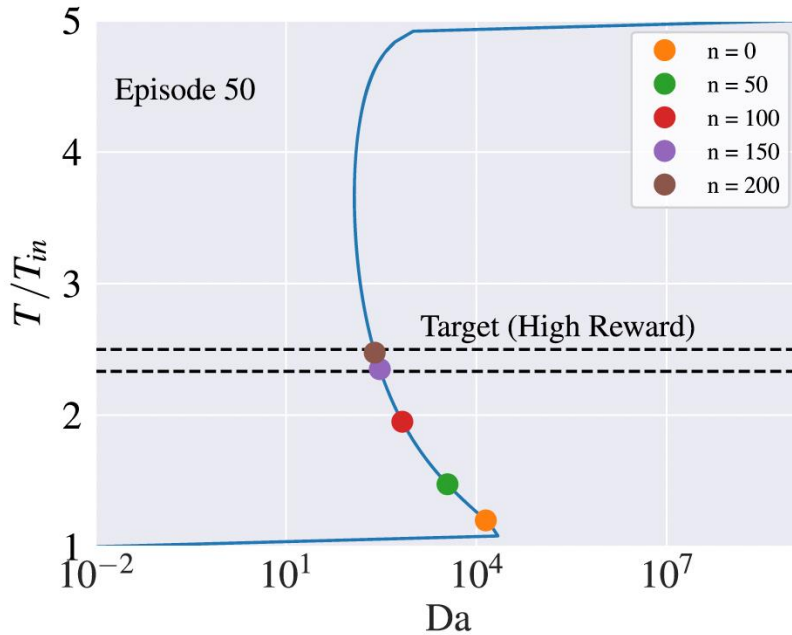
(a) Instantaneous rewards for a state-action pair.

State/Action	$T^{n+1}(Da^n) > T^n$	$T^{n+1}(Da^n) < T^n$	$T^{n+1}(Da^n) = T^n$
$T^{n+1} < T_{min}^*$	999.833	997.189	997.452
$T^{n+1} > T_{max}^*$	998.671	999.788	998.628
$T^{n+1} \in [T_{min}^*, T_{max}^*]$	999.836	999.194	999.837

(b) Q-table for $n = 600$ at the 99th episode.

RL Problem formulation

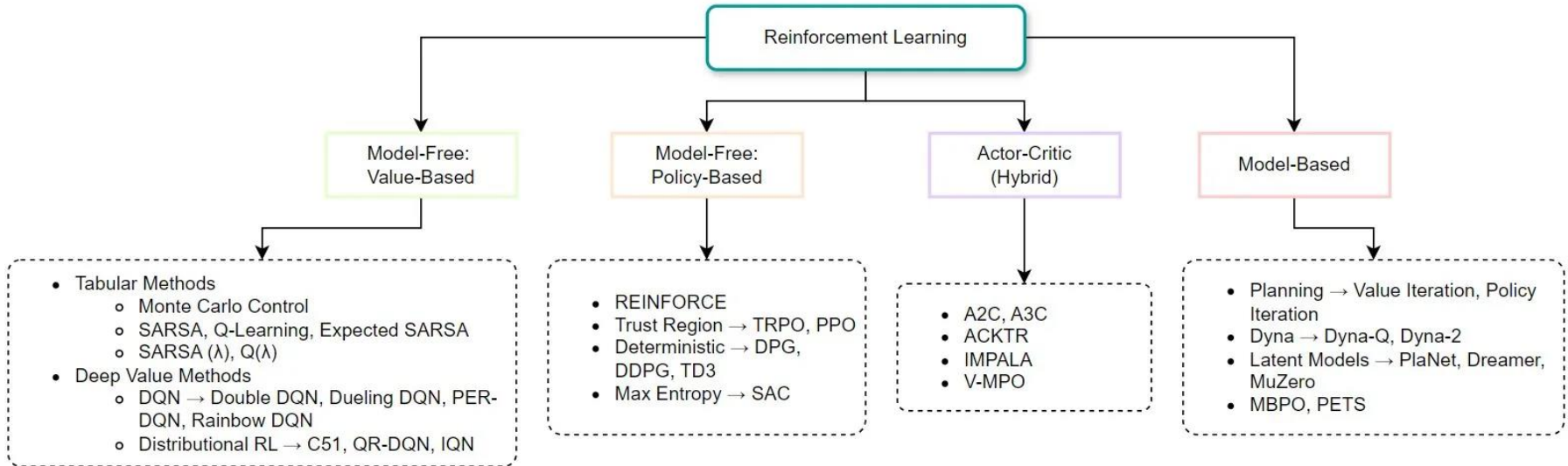
- Application example: control temperature in partially stirred reactor



RL Problem formulation

- Challenges with RL
 - Application to fully observable problems → limited sensors for combustion applications
 - Continuous vs. discrete action states
 - Data inefficiencies, required $O(10^5)$ training samples
 - Out-of-sample performance
 - Interpretability
 - Choice of reward function
- Opportunities
 - Deep learning

Reinforcement learning solution techniques



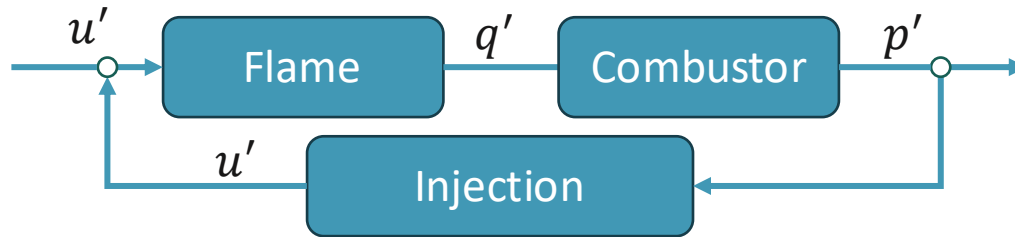
<https://medium.com/@itzcharles03/a-complete-taxonomy-of-reinforcement-learning-algorithms-from-basics-to-cutting-edge-dc51878caf77>



RL application to control thermoacoustic instabilities

What are Combustion Instabilities?

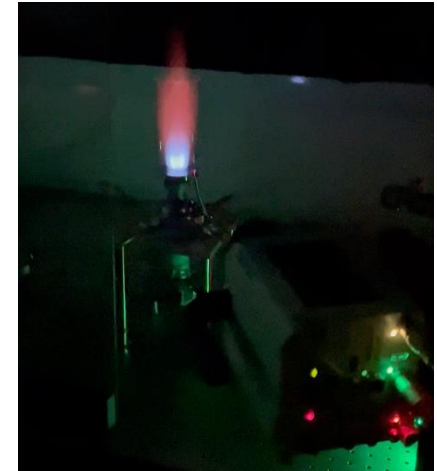
- Strong Interaction between heat release and pressure fluctuations



$$\underbrace{\nabla \cdot c_0^2 \nabla p_1}_{\text{Acoustic Perturbations}} - \underbrace{\frac{\partial^2 p_1}{\partial t^2}}_{\text{Acoustic Noise}} = \underbrace{-\gamma p_0 \nabla \vec{u} : \nabla \vec{u}}_{\text{Acoustic Noise}} - \underbrace{(\gamma - 1) \frac{\partial \dot{\omega}_T}{\partial t}}_{\text{Heat Release}}$$

Acoustic eigenmodes

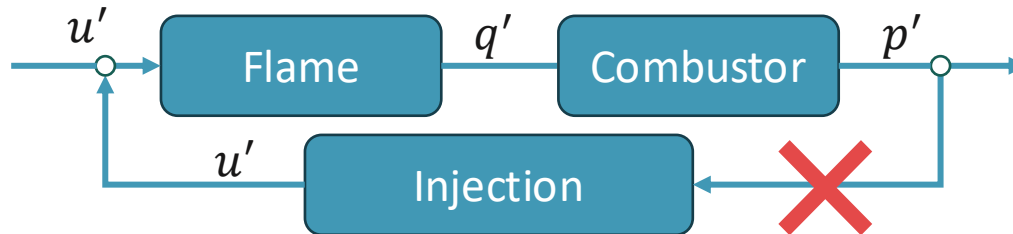
Source terms



[1] Poinsot, T., and Denis V. Theoretical and numerical combustion, 2005.

Active vs Passive Control for thermoacoustics

	Active Control	Passive Control
Definition	Observe \rightarrow control logic \rightarrow Output	No control logic
Advantages	<ul style="list-style-type: none">• Real-time adjustment of the control system• Wide range of operating conditions	<ul style="list-style-type: none">• Lack of moving component• Low maintenance cost/high durability
Drawbacks	<ul style="list-style-type: none">• Increase system complexity/cost• Require control algorithms.• Might destabilize stable system• Sometimes fails to harmonics	<ul style="list-style-type: none">• Higher expenses and timing for retrofit• Limited operating conditions
Examples	<ul style="list-style-type: none">• loudspeakers, fuel pulsations	<ul style="list-style-type: none">• Perforated plates, Helmholtz resonator

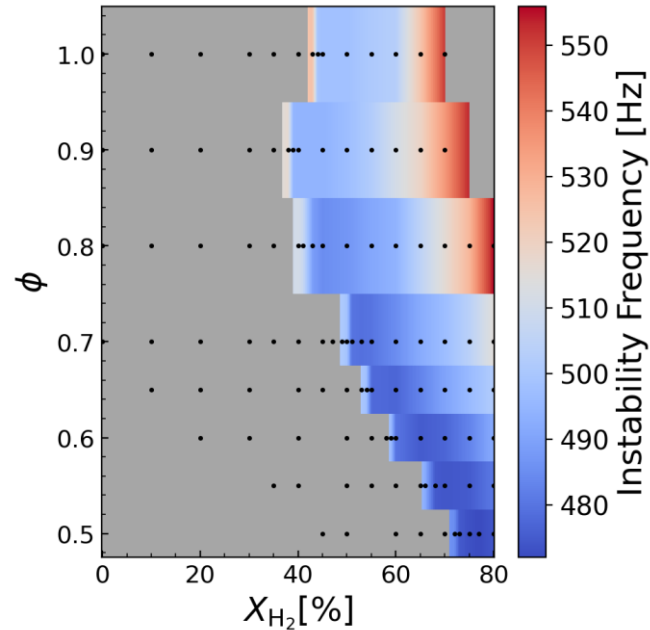
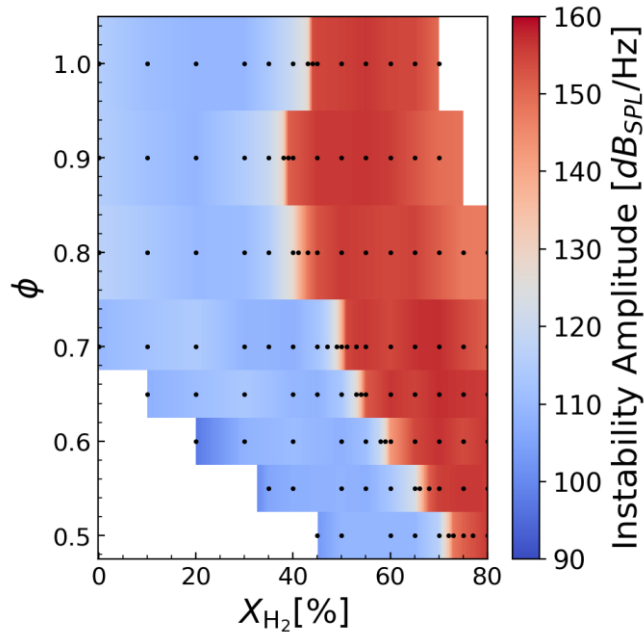


Active vs Passive Control for thermoacoustics

- Challenges in controlling thermoacoustic instabilities
 - fails to suppress all instabilities and might even destabilize a stable system
 - Requires careful tuning for each operating condition
 - Cannot easily adapt to changes in the system as it ages and degrades
- Opportunities for RL
 - Develop adaptive and flexible control strategies that can reliably suppress combustion instabilities over a wide range of conditions?
 - Real-time control of thermoacoustic instabilities
 - Can learn and adjust to complex environments in real time
 - Robust to uncertainties (experiment, sensors)
 - Improved control characteristics(adaptability, generalizability)
 - Can handle multi-objective optimization and sensor fusion

Stability Maps: ϕ, X_{H_2}

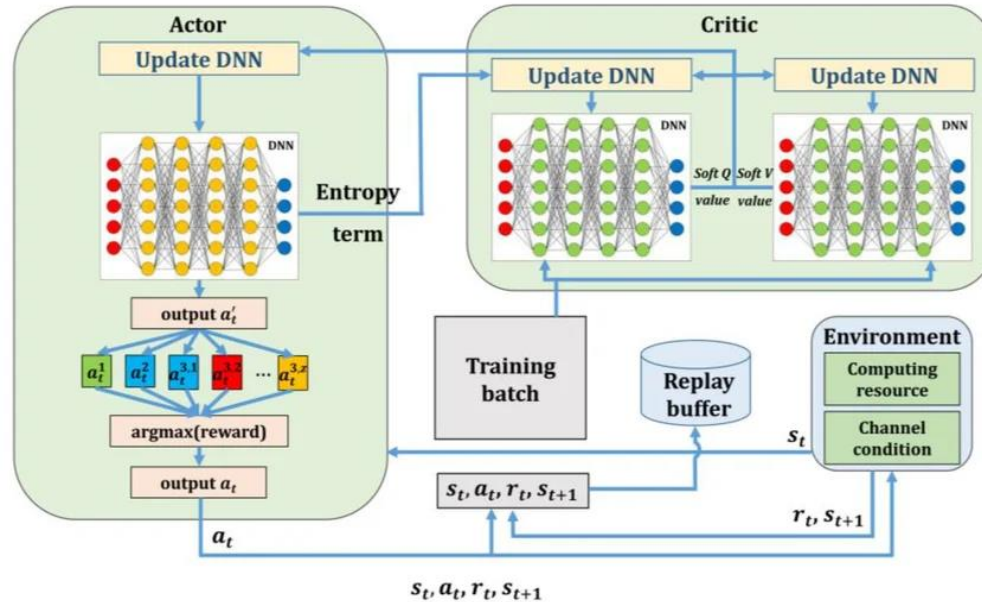
- Limit cycle amplitude of 160 dB_{SPL} at a frequency $f \approx 500$ Hz
- Reduction in instability amplitude close to $X_{H_2} \sim 80\%$



Soft-actor Critique RL

- Deep RL model for continuous action space

$$J(\pi) = \sum_{t=0}^T \mathbb{E}_{(s_t, a_t) \sim \rho_\pi} [r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot | s_t))]$$



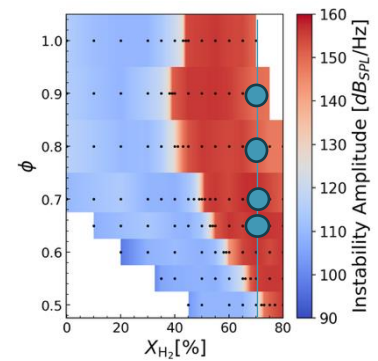
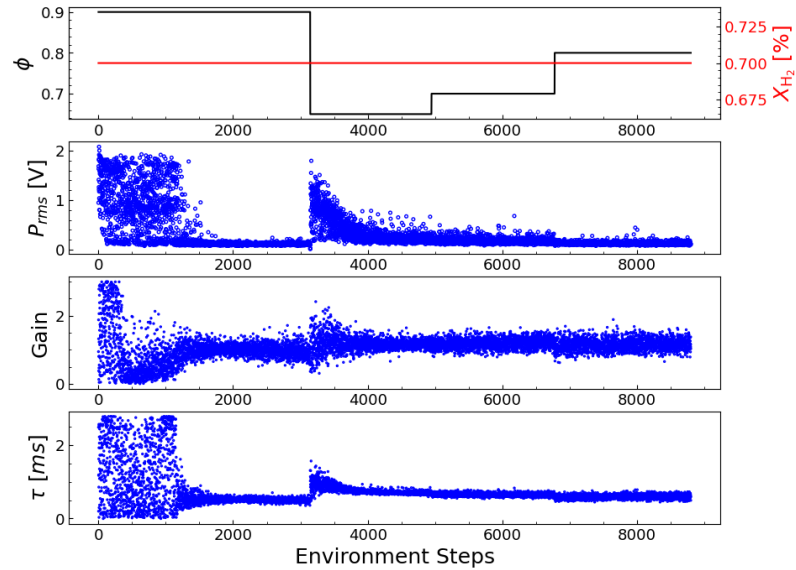
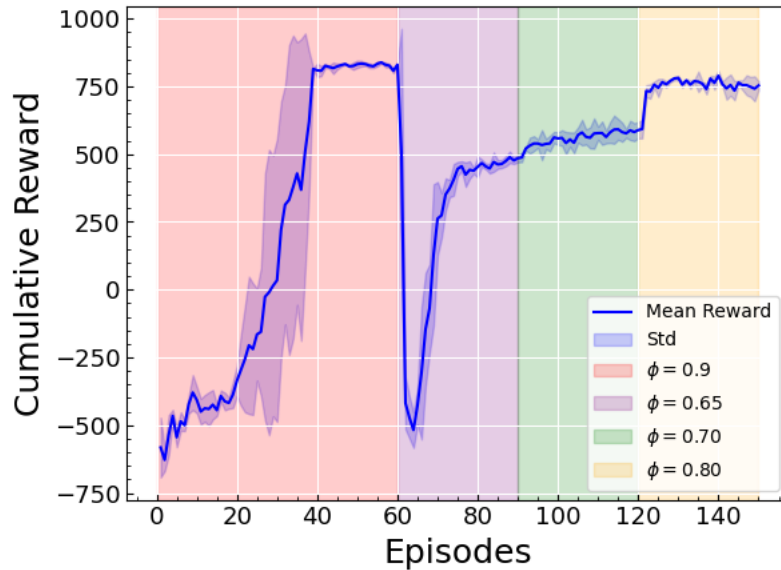
Haarnoja, T., et al, PMLR, 2018.

Soft-actor Critique RL

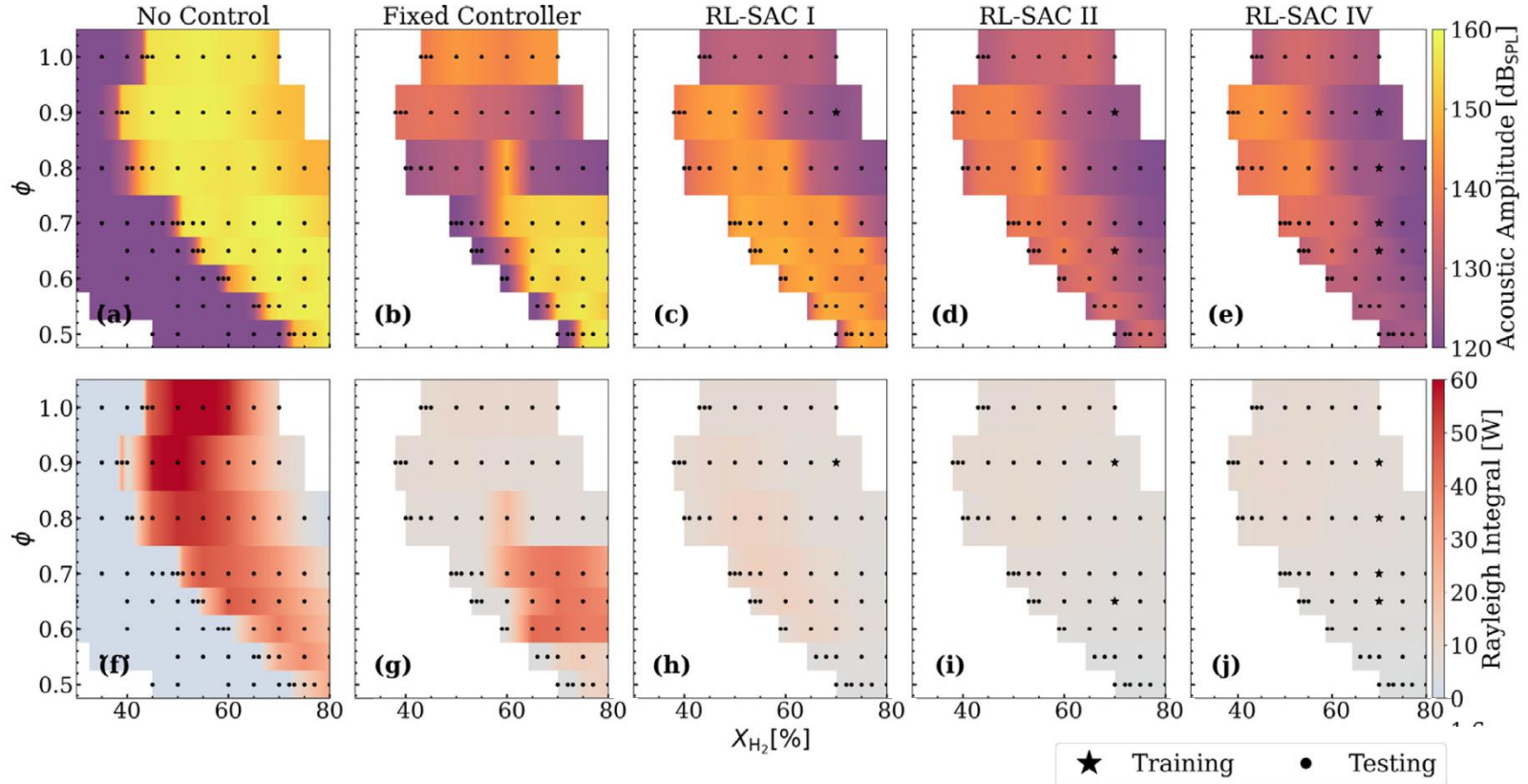
- Train neural networks
 - Actor $[\pi_\phi(s_t)]$
 - Critic $[Q_\theta(s_t, a_t)]$: estimate expected cumulative reward of taking a particular action in given state
 - Value $[V_\psi(s_t)]$: estimate expected cumulative reward from given state when following the policy π .
- Reward function
 - $r = R - p'_{rms} - 0.1(g^2 + \tau^2)$ with $R = \begin{cases} -5 & \text{if } p'_{rms} > 0.7 \\ 10 & \text{if } p'_{rms} \leq 0.7 \end{cases}$
- State Space
 - $p'_{rms}, \overline{OH^*_{rms}}, \overline{CH^*_{rms}}$
 - $\overline{OH^*}, \overline{CH^*}, \frac{\overline{CH^*}}{\overline{OH^*}}$
 - $\text{CPSD}(p', \overline{OH^*}), \theta(\text{CPSD}(p', \overline{OH^*}))$
- Action Space: g, τ

SAC – Training

- Training the agent on four conditions
 - Optimum gain and delay changes between conditions



SAC – Testing

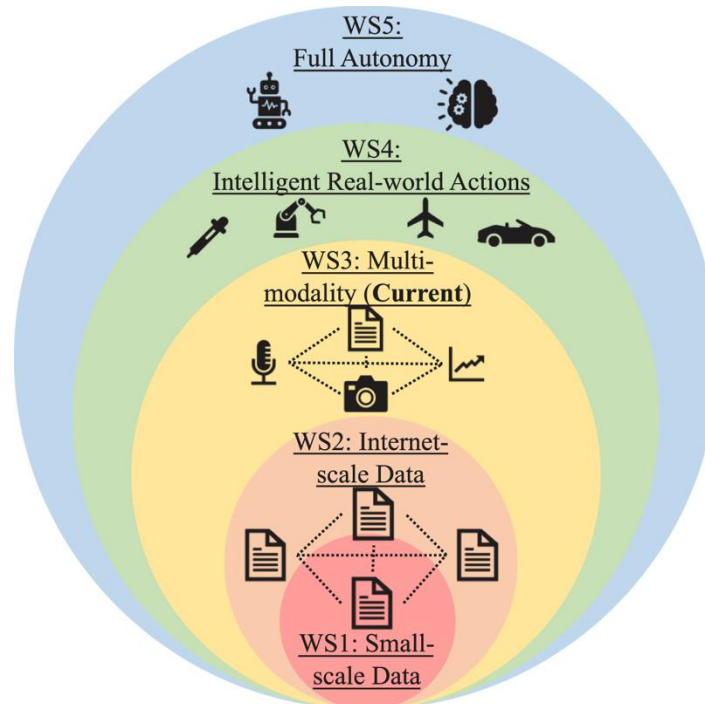




Towards Agentic AI in Scientific Discovery and Engineering

Towards Agentic AI

- Progress in ML and AI measured as world scopes (WS)

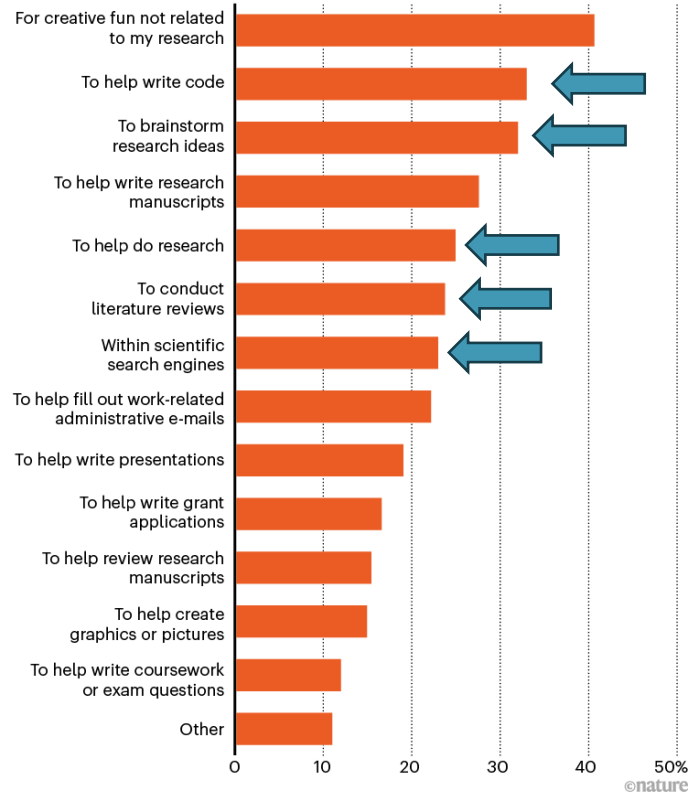


Bisk et al. Proc. Conf. Empir. Methods Nat. Lang. Process, 2020, pp. 8718–8735.

Why Are Foundation Models Important?

HOW RESEARCHERS USE LARGE LANGUAGE MODELS

Q: What do you use generative AI tools (such as ChatGPT and other large language models) for? (Choose all that apply.)



AI AND SCIENCE: WHAT 1,600 RESEARCHERS THINK

A *Nature* survey finds that scientists are concerned, as well as excited, by the increasing use of artificial-intelligence tools in research.
By Richard Van Noorden and Jeffrey M. Perkel

672 | Nature | Vol 621 | 28 September 2023

Artificial-intelligence (AI) tools are becoming increasingly common in science, and many scientists anticipate that they will soon be central to the practice of research, suggests a *Nature* survey of more than 1,600 researchers around the world.

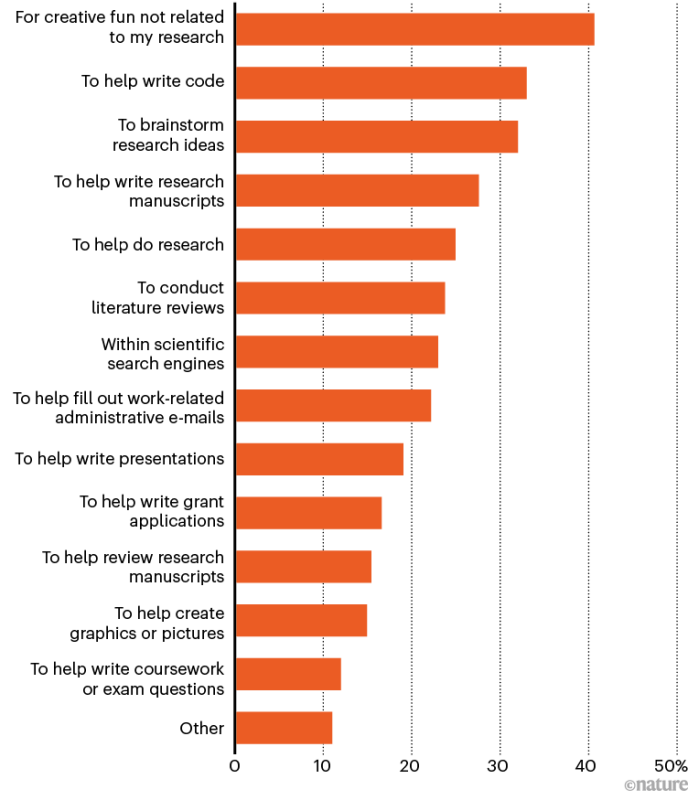
When respondents were asked how useful they thought AI tools would become for their fields in the next decade, more than half expected the tools to be 'very important' or 'essential'. But scientists also expressed strong concerns about how AI is transforming the way that research is done (see 'AI and research: survey results').

The share of research papers that mention AI terms has risen in every field over the past decade, according to an analysis for this article by *Nature*. Machine-learning statistical techniques are now well established, and the past few years have seen rapid advances in generative AI,

Why Are Foundation Models Important?

HOW RESEARCHERS USE LARGE LANGUAGE MODELS

Q: What do you use generative AI tools (such as ChatGPT and other large language models) for? (Choose all that apply.)



1. Formulating hypothesis

- Brainstorming
- Retrieving knowledge

2. Planning test

- Identify test conditions
- Develop test environments

3. Conducting tests

- Obtaining data
- Controlling experiments
- Running calculations

4. Distilling knowledge

- Processing data
- Visualizing data
- Discovering patterns

Scientific Process

How good are AI models for combustion science and engineering?

User Prompt: For the purposes of performing combustion experiments can you suggest a few studies that could be feasibly done?

AI Response:

1. Fuel Mixture Optimization for Maximum Efficiency [...]
2. Autoignition Temperature Analysis of Alternative Fuels [...]
3. Emission Characterization under Varied Operating Conditions[...]
4. Impact of Additives on Combustion Efficiency and Emissions[...]
5. Exploring Pre-ignition and Knocking Phenomena in Engines[...]
6. Thermal Decomposition of Fuels under Extreme Conditions[...]

User Prompt: ... provide me with a hypothesis ...

AI Response:

"Utilizing advanced biofuel blends with traditional hydrocarbon fuels [...] in a specific ratio can significantly improve combustion efficiency while meeting or surpassing current emission regulations."

How Good Are AI Models For Combustion Science and Engineering?

User Prompt: Write me Cantera code for investigating the combustion of different fuel blend ratios

AI Response:

Promising Biofuel Candidates

1. Ethanol: Widely used as an additive in gasoline ...
2. Biodiesel ...
- ...
8. Synthetic Biofuels ...

Sources for Chemical Mechanisms

1. Lawrence Livermore National Laboratory (LLNL) Combustion Database:
...
6. CRECK Modeling Group:
 - Based in Politecnico di Milano, they provide a range of kinetic models, including for biofuels.
 - Website: <http://creckmodeling.chem.polimi.it>
- ...

How Good Are AI Models For Combustion Science and Engineering?

User Prompt: Write me Cantera code for investigating the combustion of different fuel blend ratios

```
import cantera as ct
import numpy as np
# Define the fuel blends to test
# The ratios are given in mole fractions.
fuel_ratios = np.linspace(0, 1, 5)
# Define initial conditions
pressure = 101325 # 1 atm in Pascals
temperature = 600 # Starting temperature
# Results list
results = []

for ratio in fuel_ratios:
    # Create mixture of the two fuels [...]
    gas = ct.Solution('gri30.cti')
    # Use a generic mechanism as a base
    gas.TP = temperature, pressure
    # Set the composition based on the ratio
    ethanol_mole_fraction = ratio
    gasoline_mole_fraction = 1 - ratio
    gas.X = {'C2H5OH': ethanol_mole_fraction,
            'C8H18': gasoline_mole_fraction}
    # Create a reactor and initialize gas
    mixture
```

User Prompt: Gasoline and ethanol mechs aren't actually used here. You've reverted to GRI

How Good Are AI Models For Combustion Science and Engineering?

User Prompt: Write me Cantera code for investigating the combustion of different fuel blend ratios

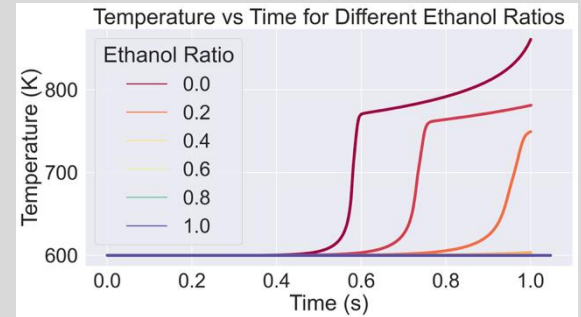
```
import cantera as ct
import numpy as np
# Define the fuel blends to test
# The ratios are given in mole fractions.
fuel_ratios = np.linspace(0, 1, 5)
# Define initial conditions
pressure = 101325 # 1 atm in Pascals
temperature = 600 # Starting temperature
# Results list
results = []
```

```
for ratio in fuel_ratios:
    # Create mixture of the two fuels [...]
    mech = 'isooctane_ethanol_mechanism.cti'
    gas = ct.Solution(mech)
    # Use a generic mechanism as a base
    gas.TP = temperature, pressure
    # Set the composition based on the ratio
    ethanol_mole_fraction = ratio
    gasoline_mole_fraction = 1 - ratio
    gas.X = {'C2H5OH': ethanol_mole_fraction,
             'C8H18': gasoline_mole_fraction}
    # Create a reactor and insert the gas
    mixture
    reactor = ct.IdealGasReactor(gas) ...
```

How Good Are AI Models For Combustion Science and Engineering?

User Prompt: Can you create code to visualize these results?

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Read the JSON file into a Pandas DataFrame
df = pd.read_json('combustion_simulation_results.json')
...plt.figure(figsize=(12, 6))
sns.lineplot(x='Time', y='T', hue='Ethanol_Ratio',
data=df)
```



User Prompt: Draw scientific conclusions based on these figures?

Here's an analysis based on the figures:...

- Higher ethanol ratios may lead to increased emissions of unburned hydrocarbons, suggesting incomplete combustion or lower combustion efficiency...

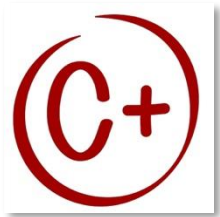
Scientific Conclusions

Emissions: Higher ethanol content may lead to increased emissions of unburned hydrocarbons, indicating a trade-off between using renewable fuel sources and achieving complete combustion.

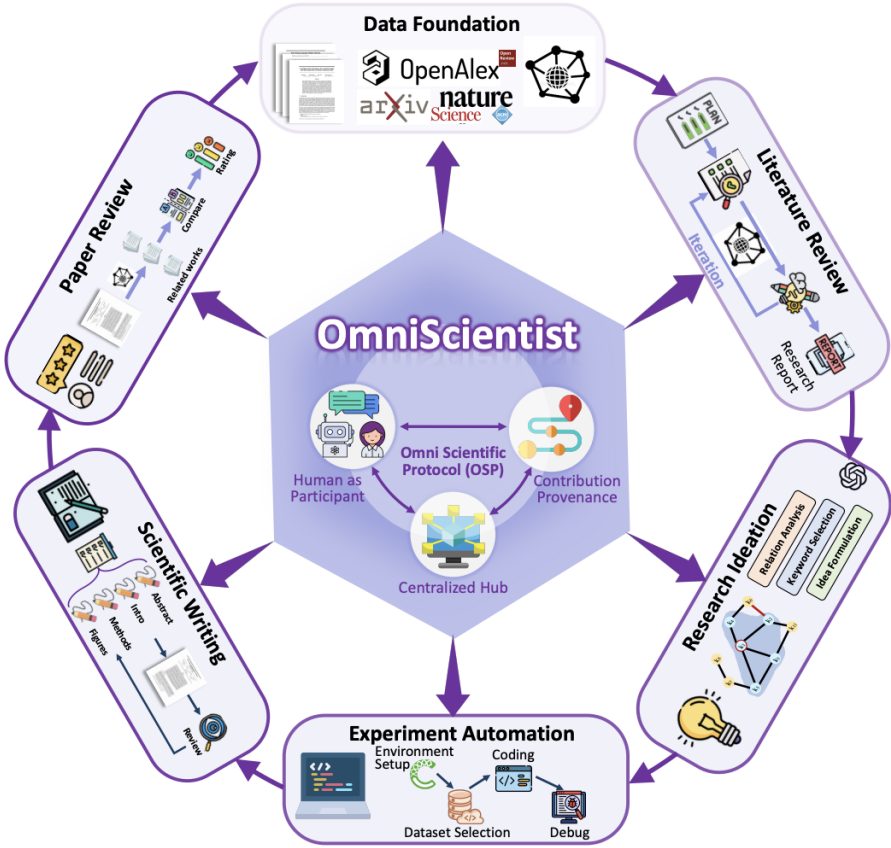
How good are LLM's for combustion tasks?

Question type	GPT-4 (2023)	Gemini 1.5 (2024)
---------------	--------------	-------------------

[Questions representative of PhD-qualifying exam]

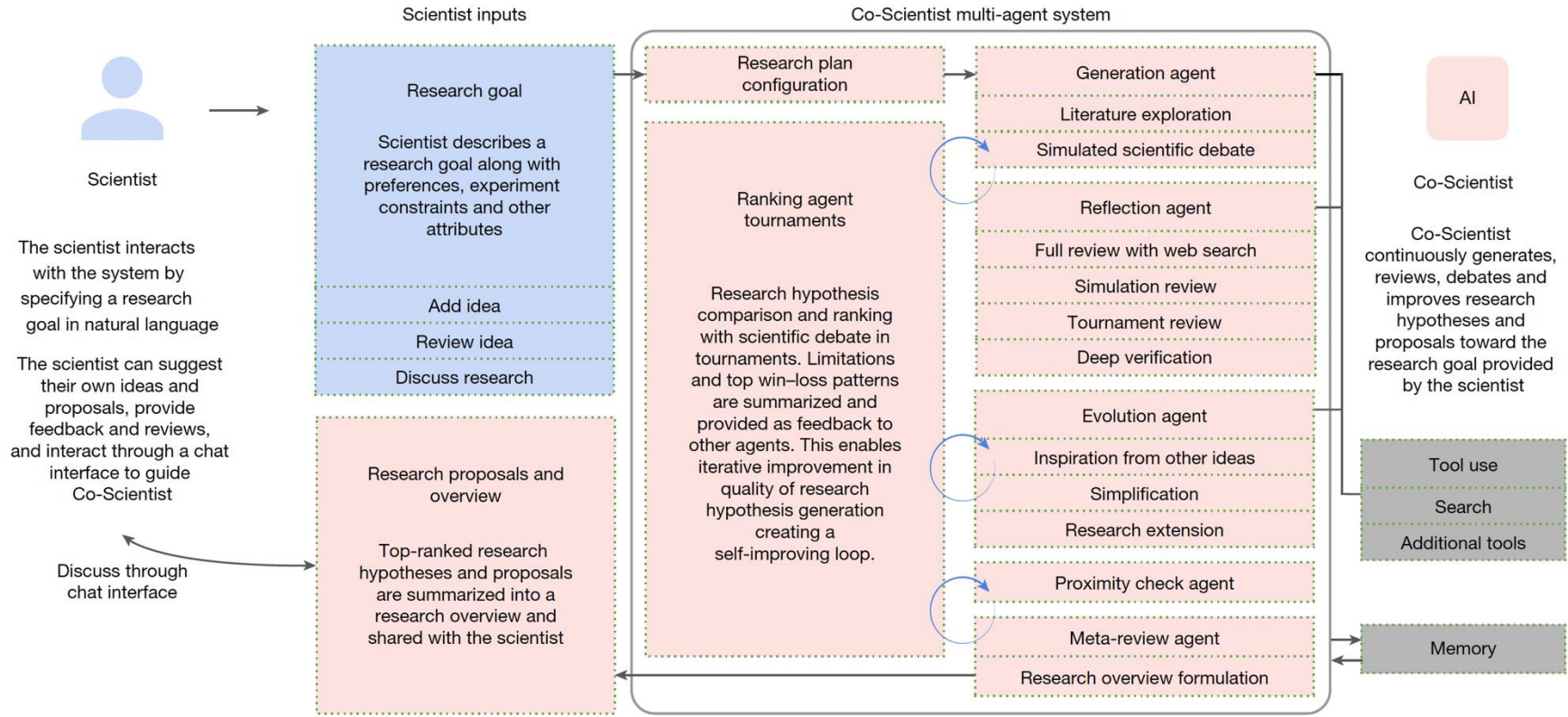


Towards agentic AI



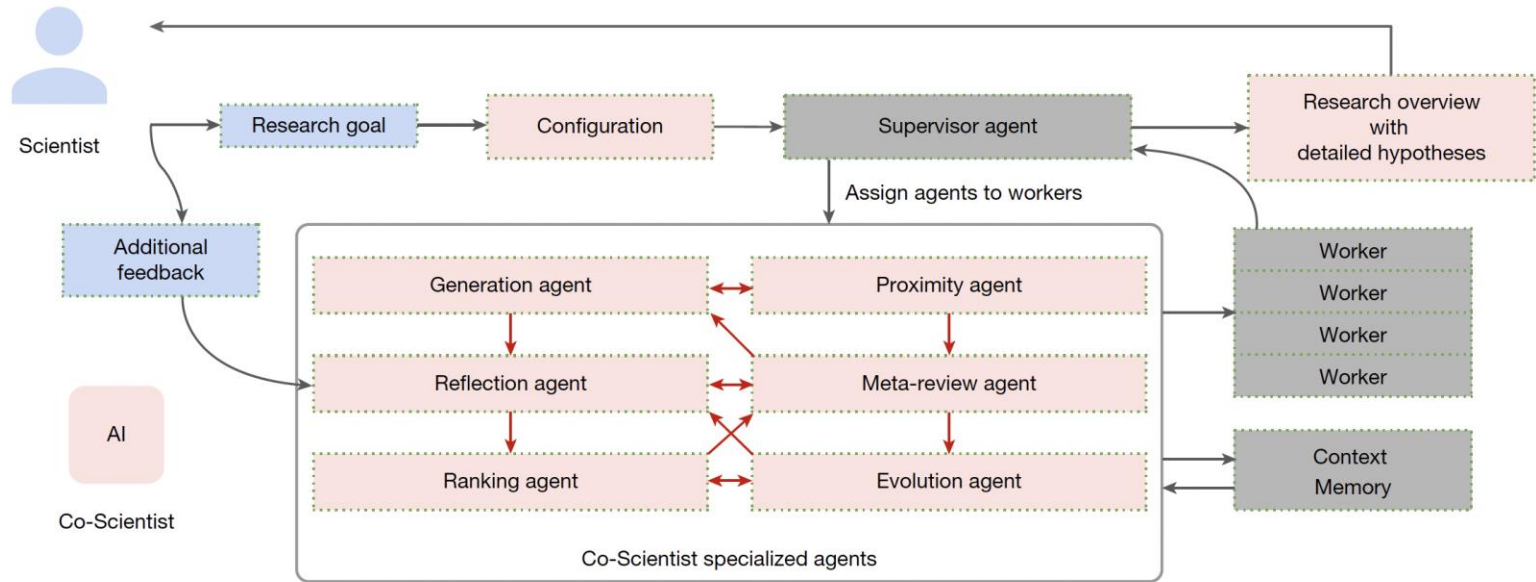
Shao et al. OmniScientist: Toward a Co-evolving Ecosystem of Human and AI Scientists, 2025

Co-Scientist

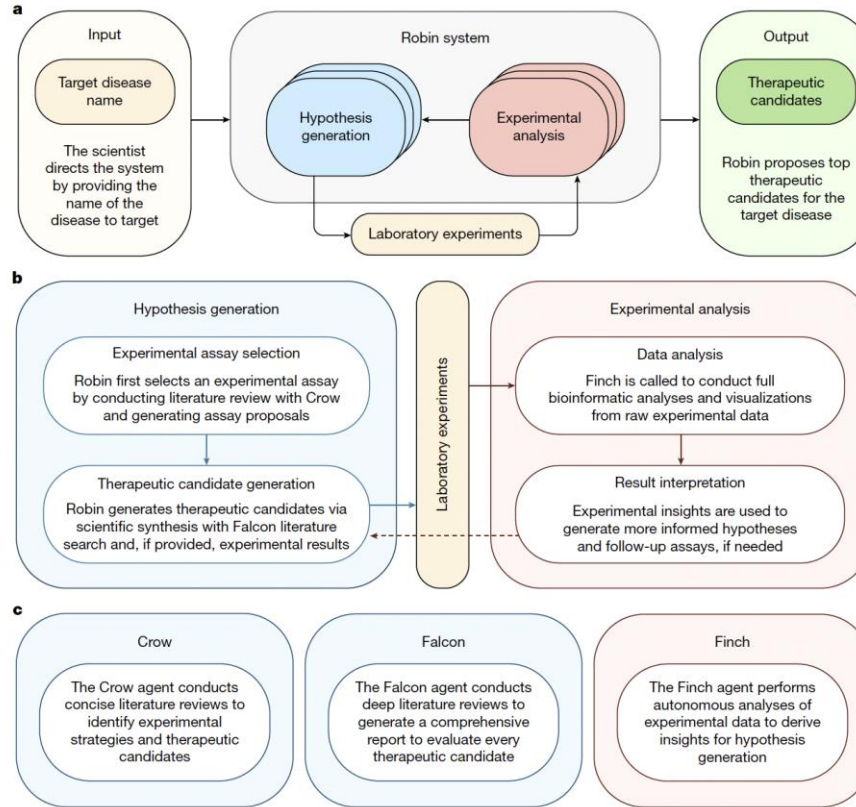


Gottweis et al., Accelerating scientific discovery with Co-Scientist, Nature, 2026

Co-Scientist



Multi-agent system for automating scientific discovery



Ghareeb et al., A multi-agent system for automating scientific discovery, Nature, 2026

Summary

- Reinforcement learning
 - Agentic method as flexible approach for data-driven control and optimization
 - Autonomous learning
 - Dynamic adaptation
 - Long-term strategic planning
 - Discovery of novel solutions
- Foundation modeling for science and engineering
- Agentic AI to accelerate scientific discovery and engineering



清华大学燃烧能源中心
Center for Combustion Energy