

# From Simulation to AI: Neural Networks for Chemical Kinetics in Turbulent Hydrogen Swirling Flames

---

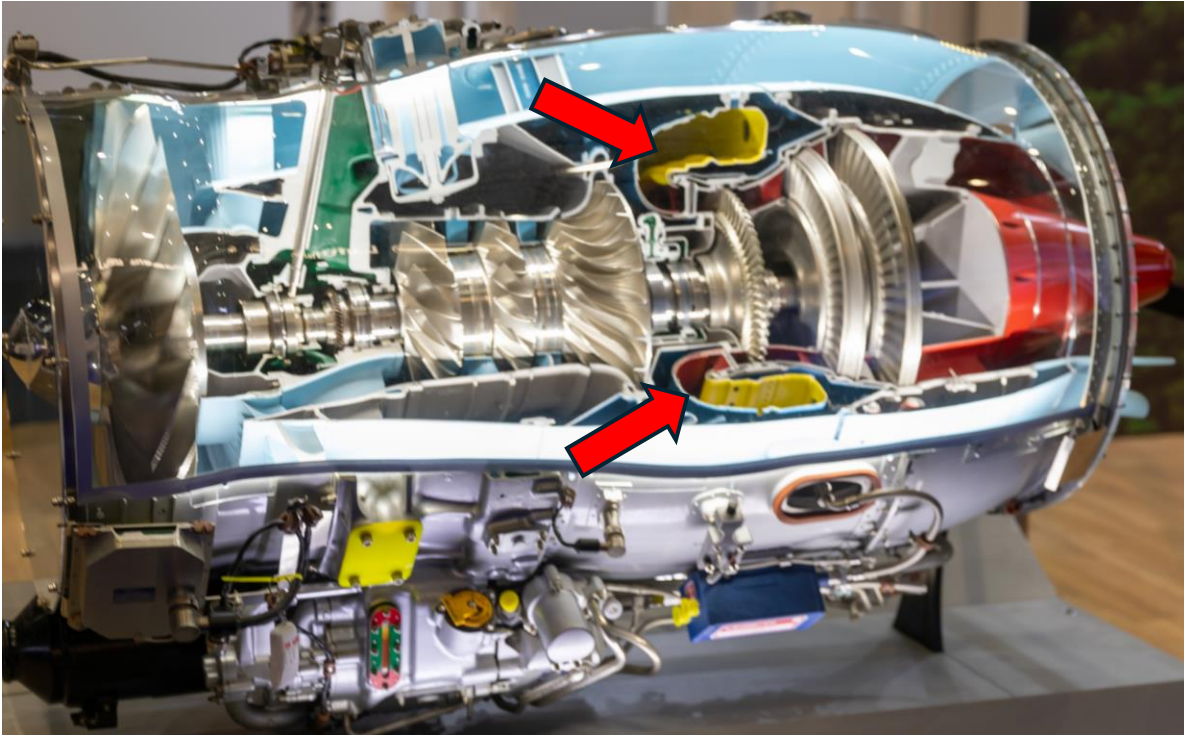
Adam Wilson

[A.Wilson43@Newcastle.ac.uk](mailto:A.Wilson43@Newcastle.ac.uk)

Newcastle University



# Why Turbulent Hydrogen Swirling Flame Chemistry Needs a Faster Model



- What is a turbulent hydrogen swirling flame?
- Hydrogen could support lower-carbon energy systems
- Direct Numerical Simulation gives exceptional detail
- A neural network surrogate could accelerate the chemistry calculation
- Assumptions

# Training Data for the Chemistry Surrogate

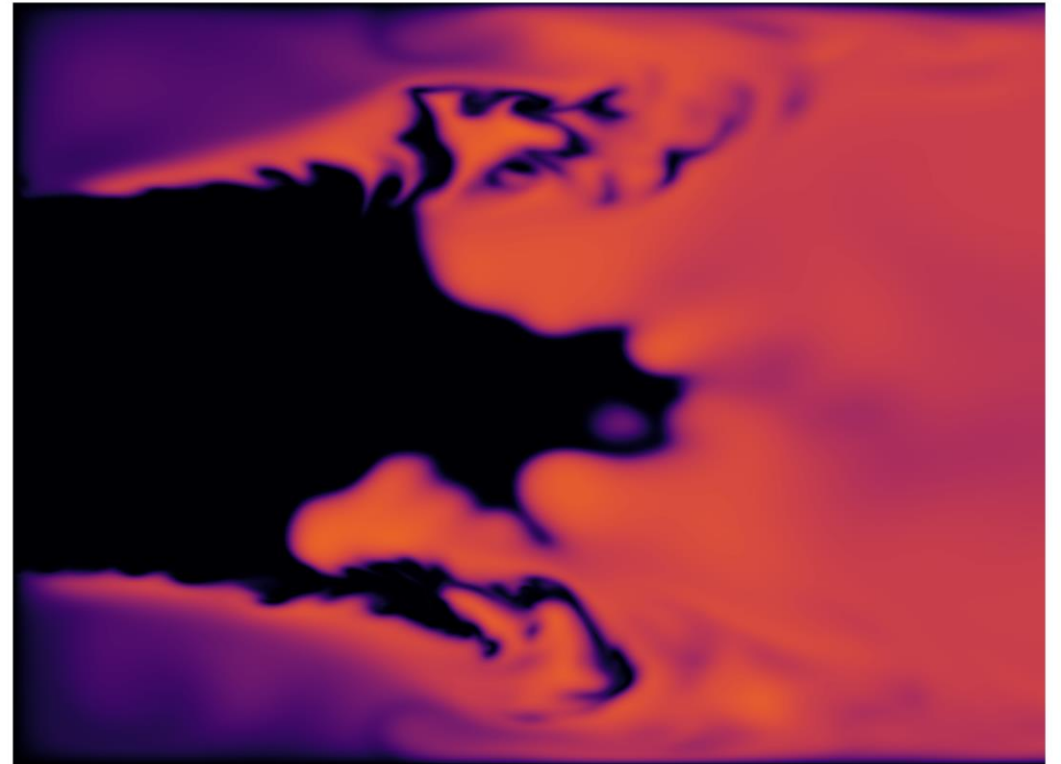
**DNS supplied realistic starting states**



1. **Quota-based sampling,**  
retaining reaction zone states
2. **Perturbed DNS states**
3. **Synthetic multi scale states**

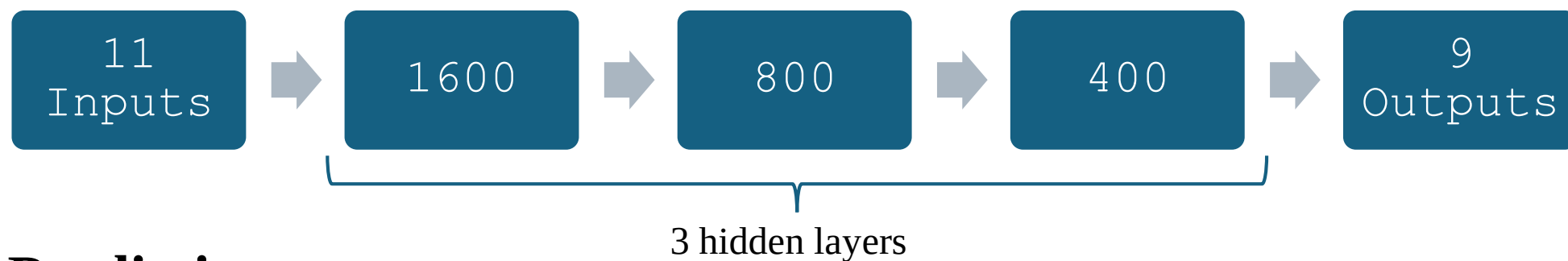


**Cantera supplied the reference answer,** creating a training pair



# Neural Network Chemistry Surrogate

1. **Input and processing**
2. **Neural Network** – fully connected feed-forward network

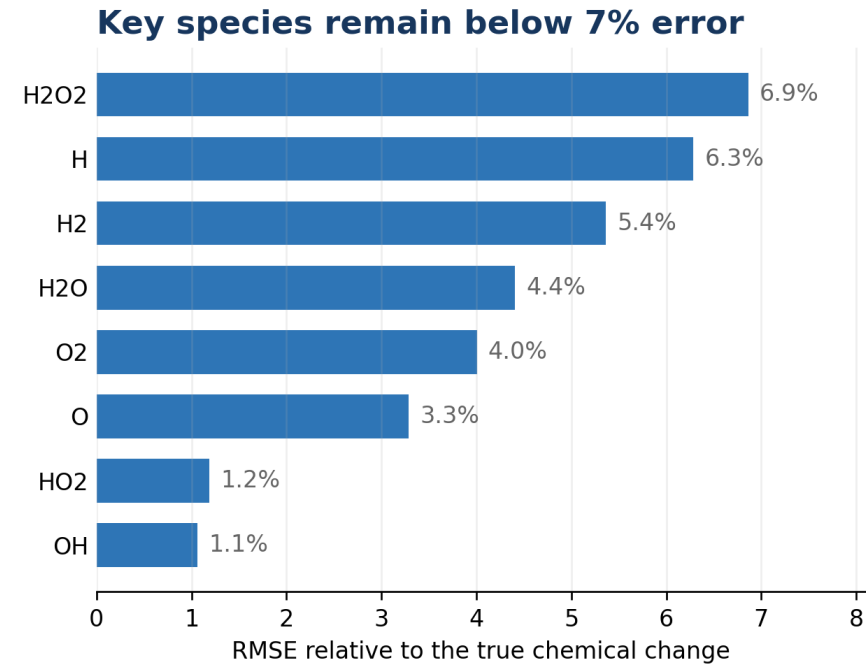
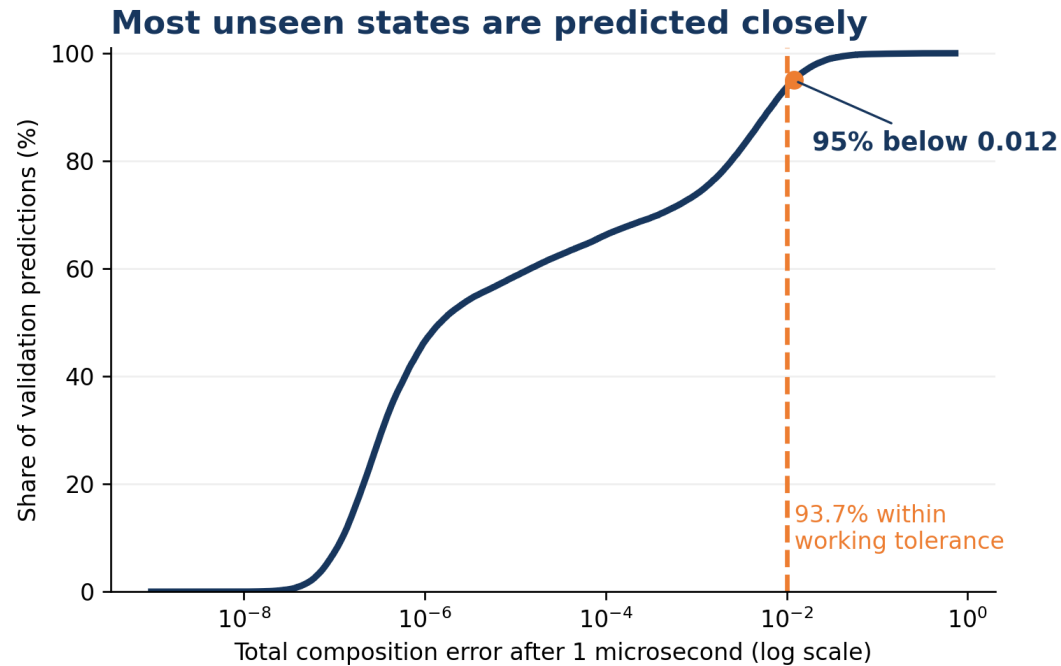


3. **Prediction**
4. **Physical correction** - hard projection

**Training** - PyTorch, Adam optimiser, 90:10 training/validation split

# Validation over 1 time step

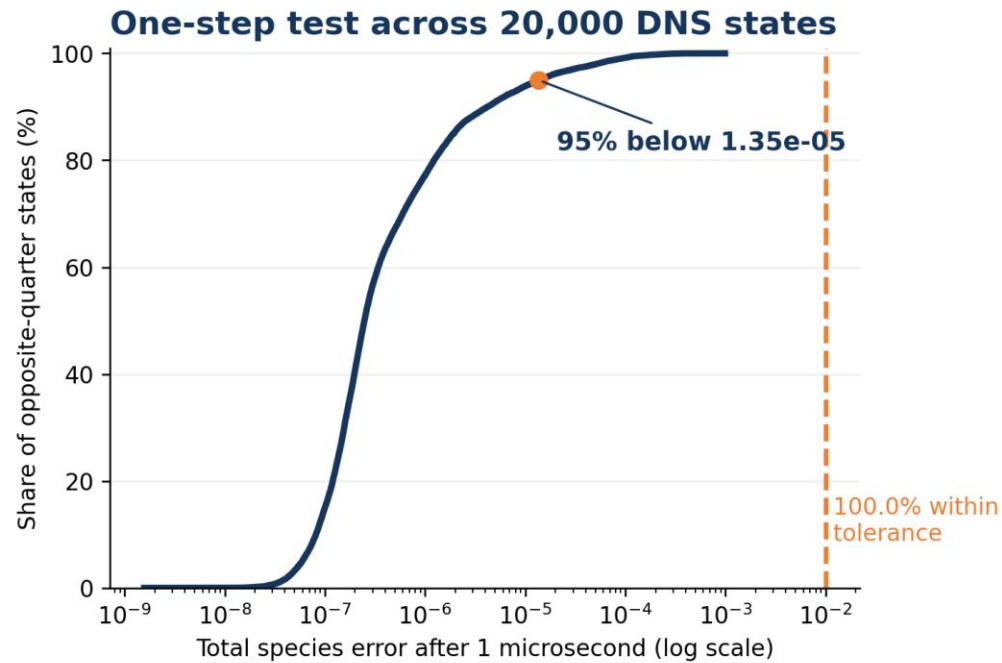
## One-step accuracy on 55,713 unseen validation states



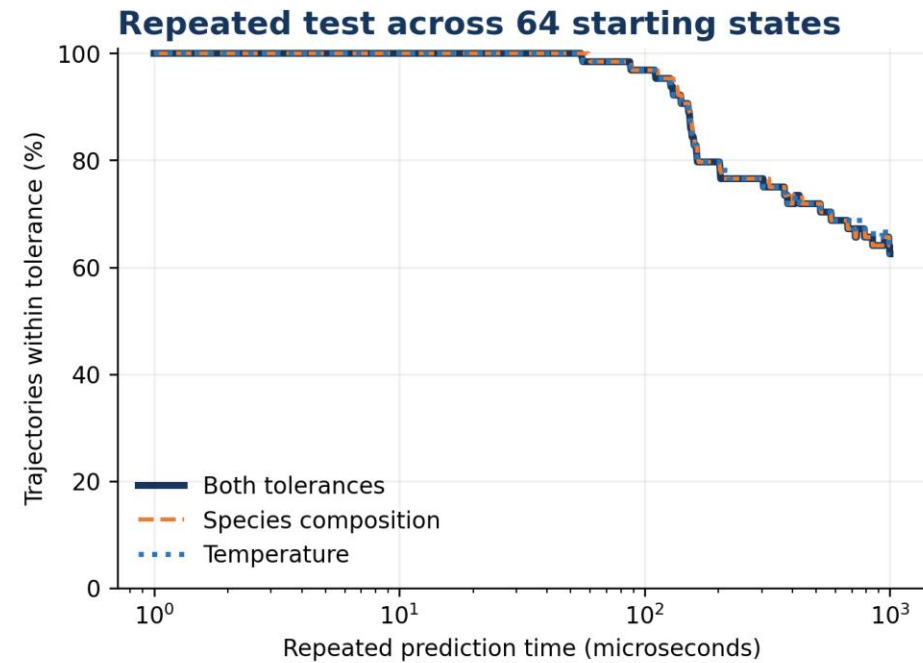
Lower is better | N<sub>2</sub> omitted from the relative-error panel because its true change is negligible

# Accuracy Over Longer Window and Unseen Data

## Opposite-quarter test: one step and repeated predictions



Training region used lower z and lower y. This test uses upper z and upper y. Recursive starting states are balanced across four temperature ranges.



# Project Outcomes and Next Steps

## **Established:**

- Trained a neural network to predict hydrogen chemistry over  $1\mu s$
- All 20,000 unseen DNS states met 1 step tolerance
- Repeated predictions accumulated error after approx.  $100\mu s$

## **Current Limitations:**

- Training using 1micro second pairs
- No testing integrated into flow solver (transport terms)
- Specific to thermochemical conditions of this dataset

## **Further Development:**

- Add physical loss term to network to become a classical PINN
- Extend time frame to allow integration into reacting flow solver

# Skills Developed through this project/Acknowledgments

- **Scientific Computing**
  - Python, PyTorch, Cantera, HDF5
- **High Performance Computing**
  - Linux, Slurm, parallel processing on Comet
- **Machine Learning**
  - Dataset design, Neural Network training and validation
- **Physics-informed Modelling**
- **Research Practice**
  - Independent problem-solving, communicating technical results

Acknowledgements:

Thank you to my supervisors,

**Umair Ahmed and Xinyi Chen**

With thanks to

