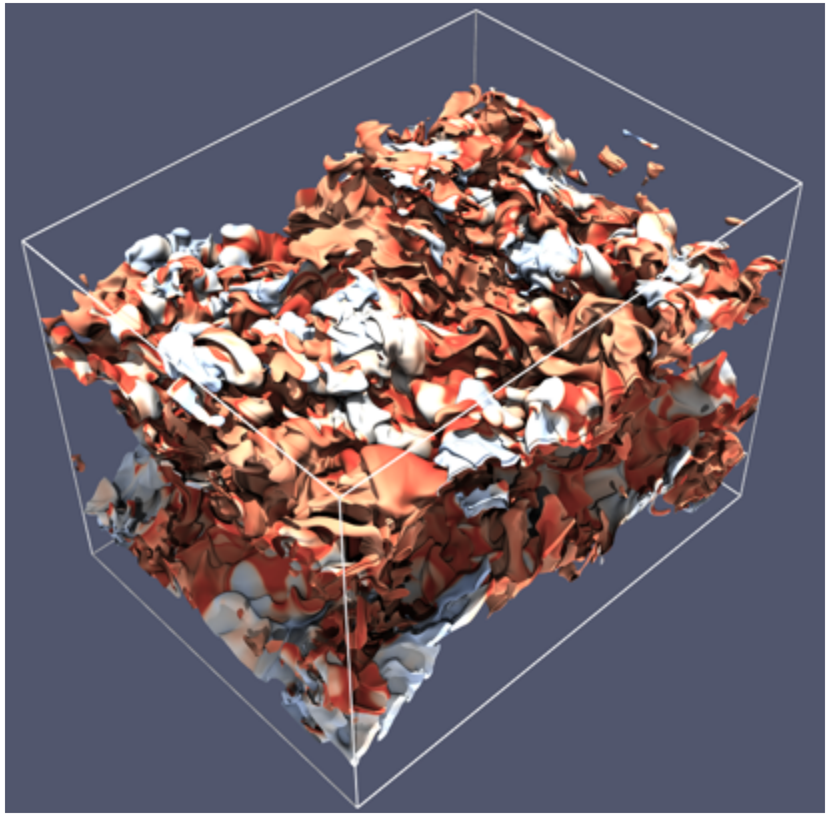
Sub-grid Scale Modeling at Scale with Deep Learning and up to 60 Billion Degrees of Freedom



M. Bode[1], D. Denker,[1] J. Jitsev[2], H. Pitsch[1]

[1]Institute for Combustion Technology
RWTH Aachen University

[2]Institute for Advanced Simulation
Juelich Supercomputing Center

Acknowledgement

- Thanks for assistance: Zeyu Lian, Michael Gauding, Marco Davidovic, Lukas Berger
- Thanks for funding by ERC “MILESTONE“
- Thanks for computing time on JURECA (JHPC55) and JUWELS (JHPC09)

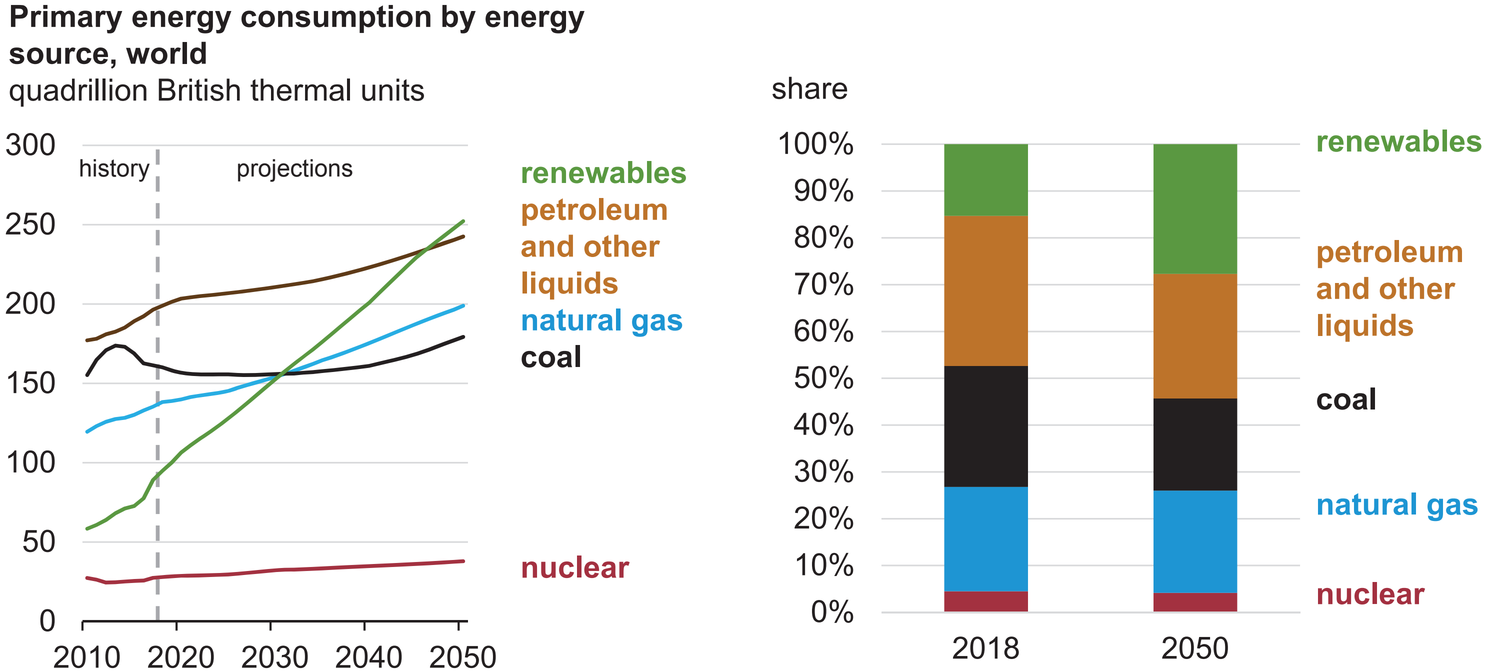
US DOE's International Energy Outlook 2019

World Energy Consumption

- Increase in world wide energy consumption
  from 2018 until 2050: 50%
- Fossil fuels > 70% by 2050

Large numbers

- 120 million tons CO_2 emissions daily in 2040
  - 13 kg per person daily
- 10 billion liter daily fuel consumption
  - 1.3 liter liquid fuel use daily



- Source: EIA's International Energy Outlook, 2019

Introducing New Renewable Fuels

Opportunity: Fuel Design

- Biofuels, E-fuels → Biohybrid fuels
- Design fuel molecules for optimized behavior
  - High efficiency by tailored reactivity
  - Low emissions

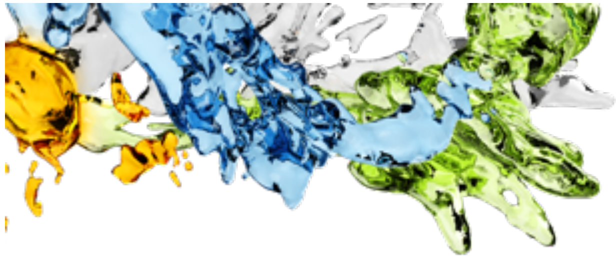
Challenge: Engine/Fuel Compatibility

- Different properties
  - Injection system needs to redesigned
  - Combustion process needs to be redesigned
- Joint optimization process of engine and fuel

→ **Quantitative, accurate, fast models relating fuel structure to performance criteria**

Tailor-Made Fuels from Biomass

The Fuel Science Center



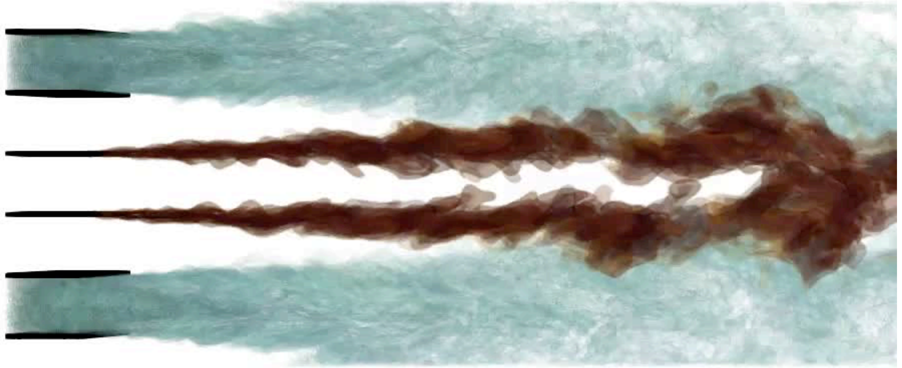
itv | RWTH AACHEN UNIVERSITY

Motivation

Turbulent Combustion

- Simultaneous optimization of efficiency, emissions and combustion stability
- New technologies:
  - Aircraft engines
    - Lean direct injection (LDI)
  - Internal Combustion Engines
    - Homogeneous charge compression ignition, Controlled auto-ignition (HCCI, CAI)
    - Downsizing with supercharging
  - Power Generation
    - Oxy-Combustion
    - Integrated gasification combined cycle (IGCC)
    - Flameless Oxidation (FLOX)

Technology development fundamentally relies on a good understanding of turbulence and of turbulent combustion.

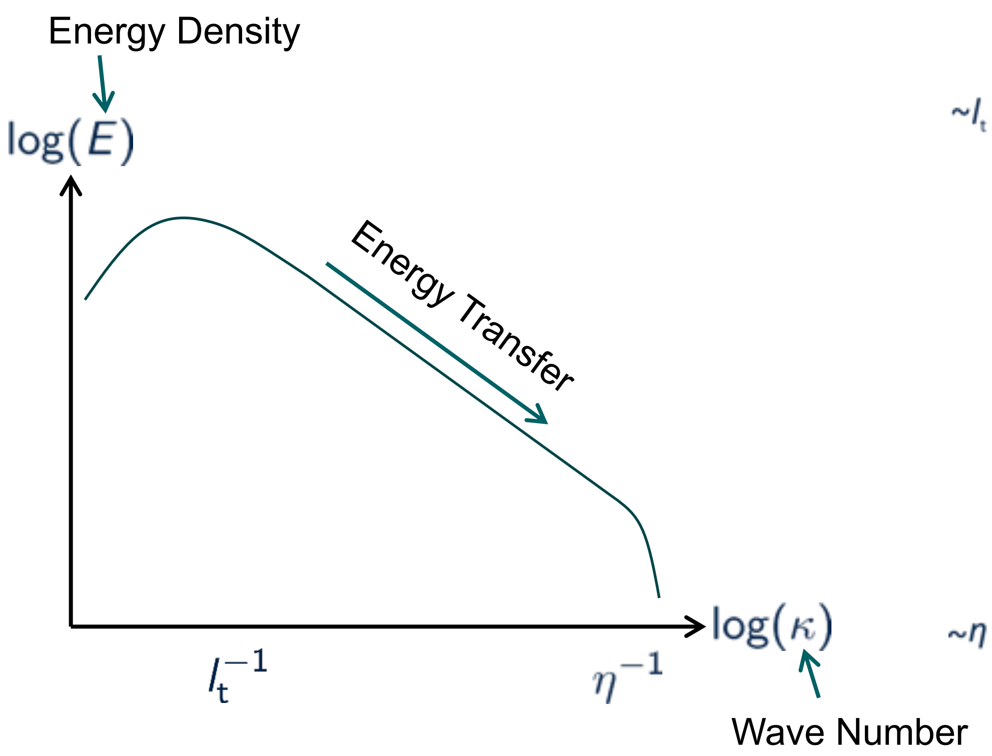


Turbulence – a very Brief Introduction

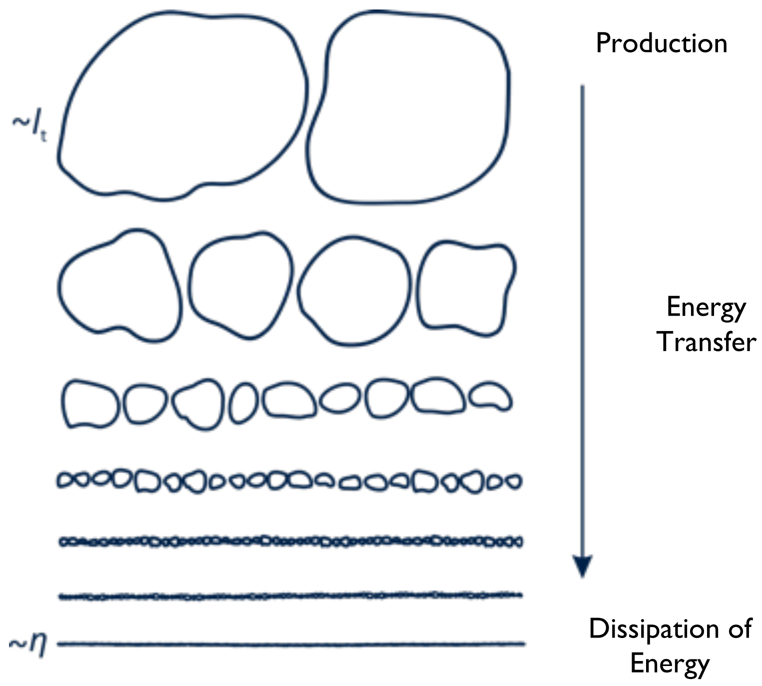
Characteristics of Turbulence:

1. Randomness
2. Multi-Scale
3. Non-Linear
4. Three-Dimensionality
5. Vorticity
6. Non-Gaussian
7. Non-Local

Energy Spectrum



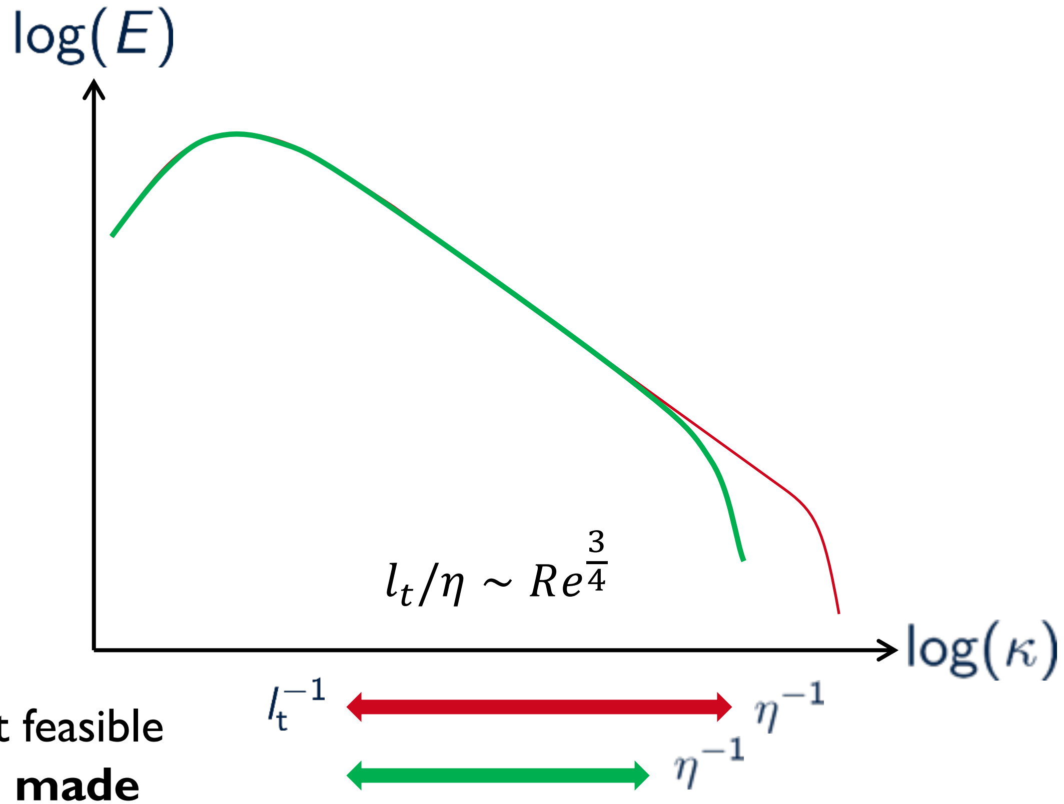
Energy Cascade



Direct Numerical Simulations

- **Problem:** lack of analytic results in turbulence research
- Two approaches:

1. Experiments
   - \+ large Reynolds numbers Re achievable
   - \- difficult to obtain full 3D fields of large fluid volumes
   - \- only indirect / impossible measurement of important quantities
2. Direct Numerical Simulations (**DNS**)
   Solving the full Navier-Stokes equations for all physically relevant scales.
   - \+ directly obtaining all relevant quantities
   - \+ perfect control of initial and boundary conditions
   - \- very high computational costs



Reynolds numbers encountered in engineering applications not feasible

- Concessions to the numerical setup must be made

Turbulent Mixing

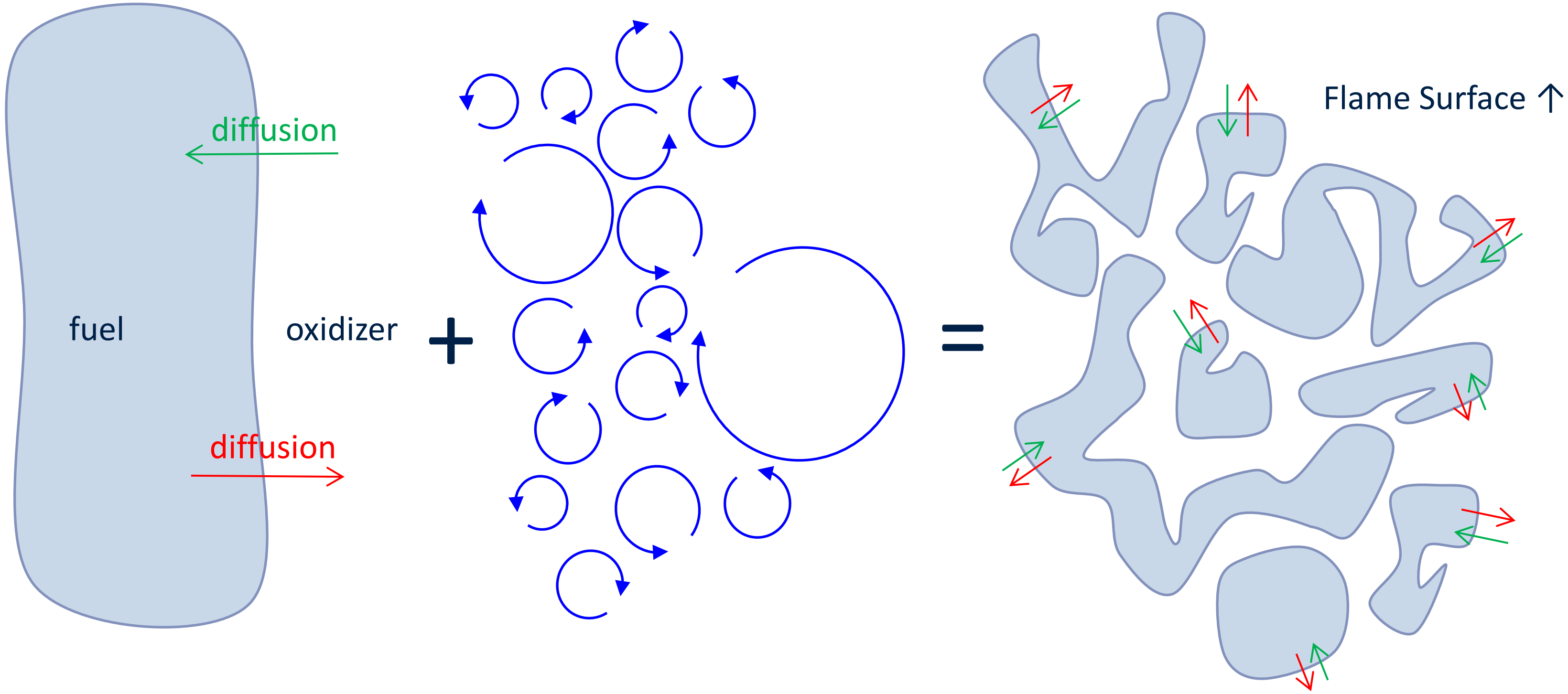
What makes turbulence important for combustion?

- Prerequisite for combustion: molecular mixing of fuel and oxidizer.
- Turbulence: added advective transport greatly enhances molecular mixing.
- Fun facts:

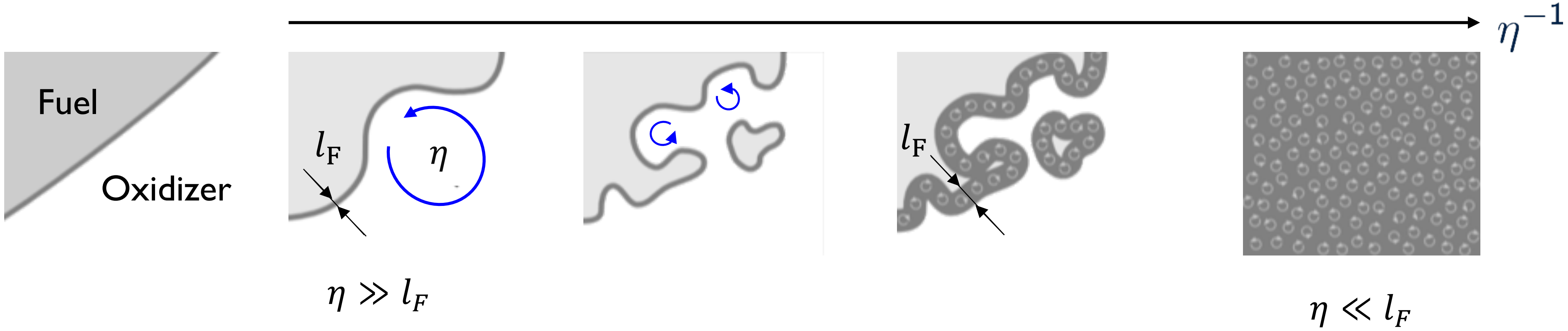
  Without turbulent mixing,

  – combustors in aircraft engines would exceed **100m** in length,

  – Passenger car internal combustion engines would be limited to **500** rpm.



Scale Interaction Between Turbulence and Combustion



Adequate combustion model exists: ✓ ✓ (✓) ✗ ✓

New advanced combustion technologies rely on dilutions of either fuel or oxidizer. Consequently, l_F increases and the combustion takes place in conditions that are not well understood.

➡ designing DNS specifically for these conditions

DNS of Reacting Flows

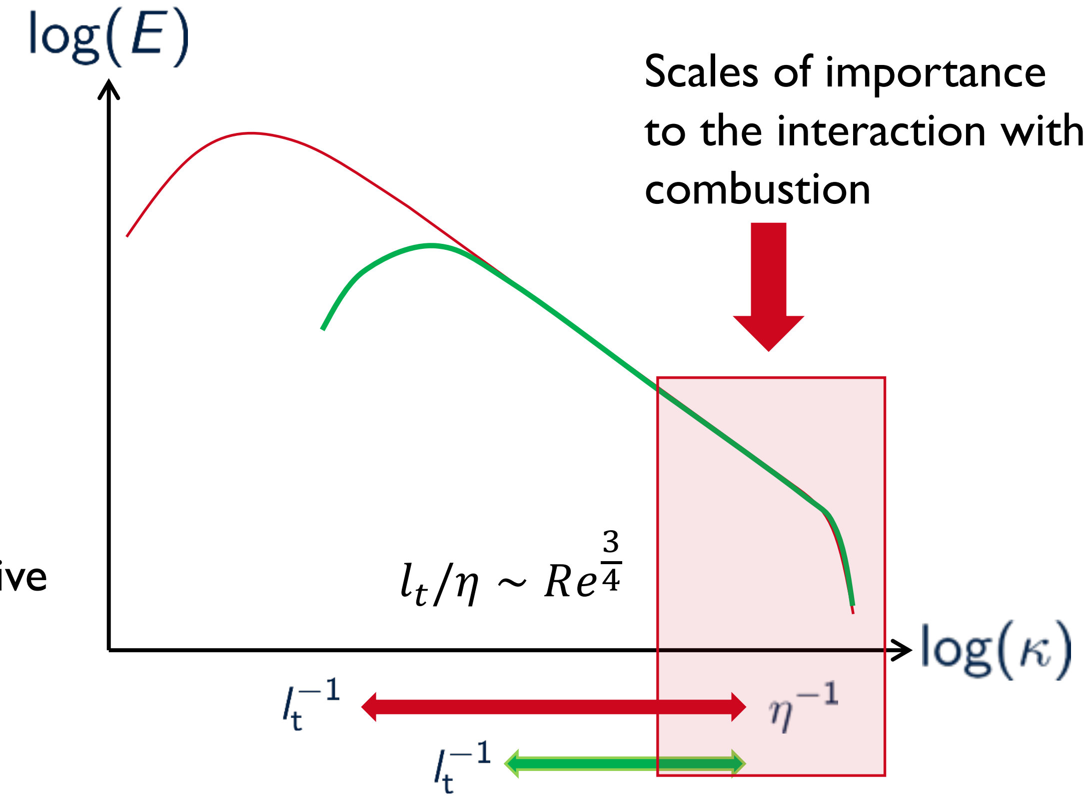
- Direct Numerical Simulations of reacting flows

- All **flow scales** need to be resolved:
  - Domain size needs to extend several l_t to capture large scale flow characteristics
  - Computational grid needs to be fine enough to resolve η
- All **flames scales** need to be resolved:
  - Simplified chemical mechanism must capture important features such as extinction and re-ignition
  - Reaction layers must be spatially resolved at all times

$$\underbrace{\hspace{10em}}$$

Reacting DNS more than an order of magnitude more expensive than non-reacting DNS of similar Reynolds number

→ l_t significantly **smaller** than in real world engineering applications.



Governing Equations – Numerical Methods

- Using the in house developed flow solver **CIAO** to solve the reacting Navier-Stokes equations in the low-Mach limit.

Continuity:
$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_\beta}(\rho u_\beta) = 0,$$

Momentum:
$$\frac{\partial \rho u_\alpha}{\partial t} + \frac{\partial}{\partial x_\beta}(\rho u_\alpha u_\beta) = -\frac{\boxed{\partial \Pi}}{\partial x_\alpha} + \frac{\partial \tau_{\alpha\beta}}{\partial x_\beta},$$

Species:
$$\frac{\partial \rho Y_i}{\partial t} + \frac{\partial}{\partial x_\alpha}(\rho(u_\alpha + V_{\alpha,i}) Y_i) = \dot{m}_i,$$

Temperature:
$$\frac{\partial \rho c_p T}{\partial t} + \frac{\partial}{\partial x_\alpha}(u_\alpha \rho c_p T) = \frac{\partial}{\partial x_\alpha}\left(\lambda \frac{\partial T}{\partial x_\alpha}\right) + \rho \frac{\partial T}{\partial x_\alpha}\sum_{i=1}^{n} c_{p,i} Y_i V_{i,\alpha} + \sum_{i=1}^{n} h_i \dot{m}_i + \dot{q}_R.$$

Split of Strang

$$\mathcal{F}_{dt}^{\mathrm{C}} : \begin{cases} \frac{\partial \rho Y_i}{\partial t} = \dot{m}_i \\ \frac{\partial \rho c_p T}{\partial t} = \sum_{i=1}^{n} h_i \dot{m}_i. \end{cases}$$

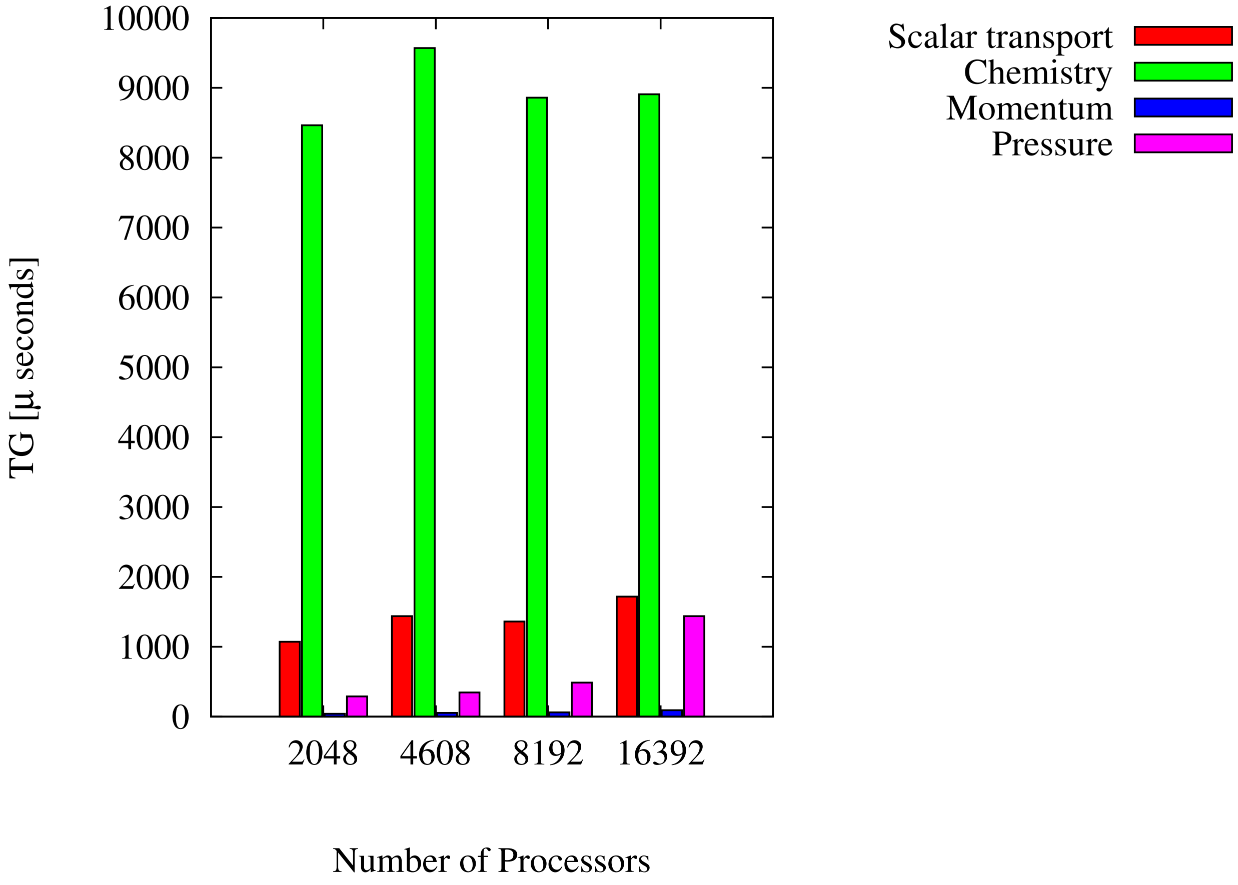
$$\mathcal{F}_{dt}^{\mathrm{Trans}} : \begin{cases} \frac{\partial \rho Y_i}{\partial t} + \frac{\partial}{\partial x_\alpha}(\rho(u_\alpha + V_{\alpha,i}) Y_i) = 0 \\ \frac{\partial \rho c_p T}{\partial t} + \frac{\partial}{\partial x_\alpha}(u_\alpha \rho c_p T) = \frac{\partial}{\partial x_\alpha}\left(\lambda \frac{\partial T}{\partial x_\alpha}\right) + \rho \frac{\partial T}{\partial x_\alpha}\sum_{i=1}^{n} c_{p,i} Y_i V_{i,\alpha} + \dot{q}_R. \end{cases}$$

$$\mathcal{F}_{dt}(Y_i^m, T^m) = \mathcal{F}_{dt/2}^{\mathrm{Trans}} \mathcal{F}_{dt}^{\mathrm{C}} \mathcal{F}_{dt/2}^{\mathrm{Trans}} \longrightarrow (Y_i^{m+1}, T^{m+1})$$

- Crank-Nicolson time advancement
- Fourth order accurate finite differences
- Poisson equation for the pressure solved with HYPRE – AMG
- Species and temperature eqs. discretized with fifth order WENO
- Chemistry ODE solved with Sundials CVODE

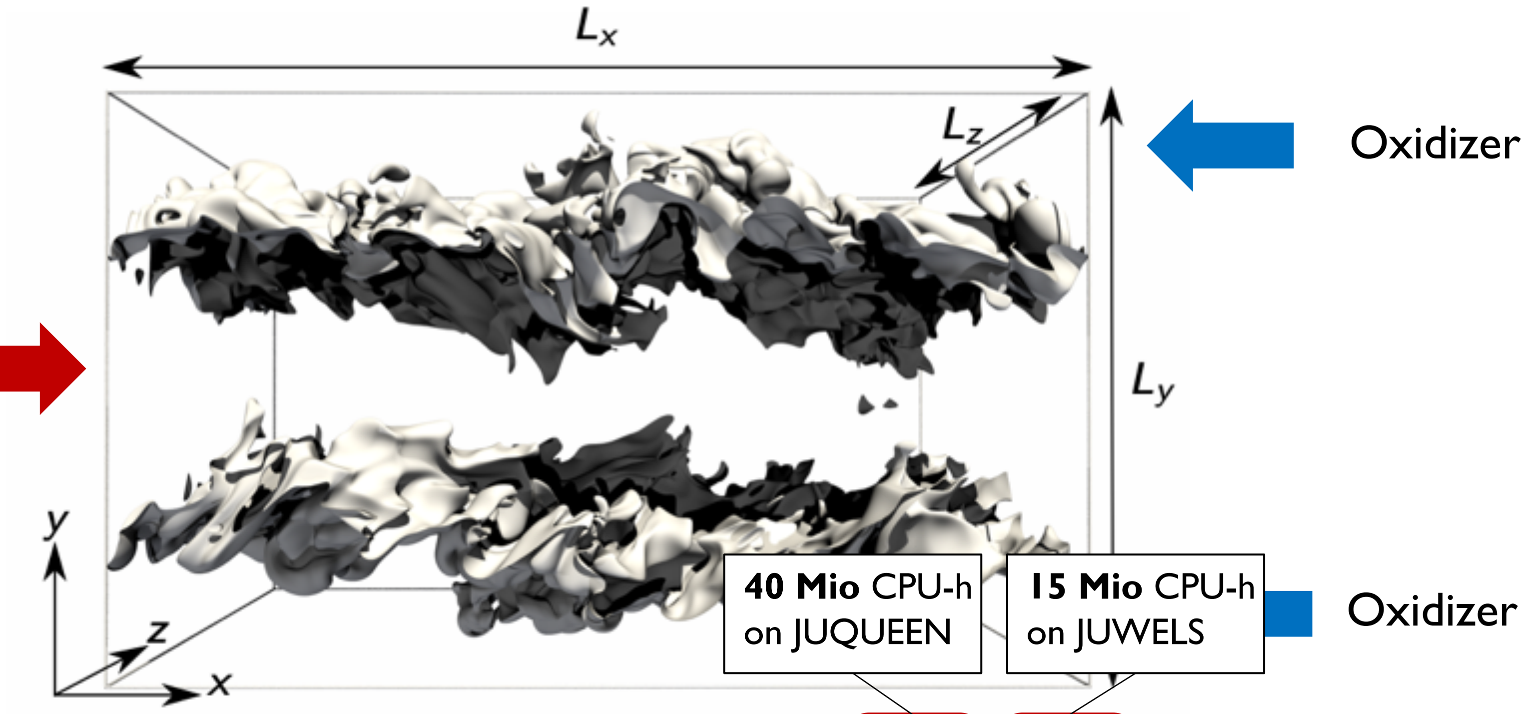
Split of Computational Costs

- Computational costs of reacting DNS
- TG – simulation time per time step and grid point
- The two most computationally expensive steps:
  - Chemistry
  - Scalar Transport
- Solving the Poisson equation is significantly more expensive than in constant density flows.



DNS of Non-Premixed Jet Flames

- Configuration:
  planar temporally evolving jet.
- Advantages:
  - maximized flame surface
  - ease of obtaining statistics
- Chemistry included via Finite-rate chemistry
- Chemical Mechanism features 30 species and 102 reactions
- Fuel: highly diluted methane

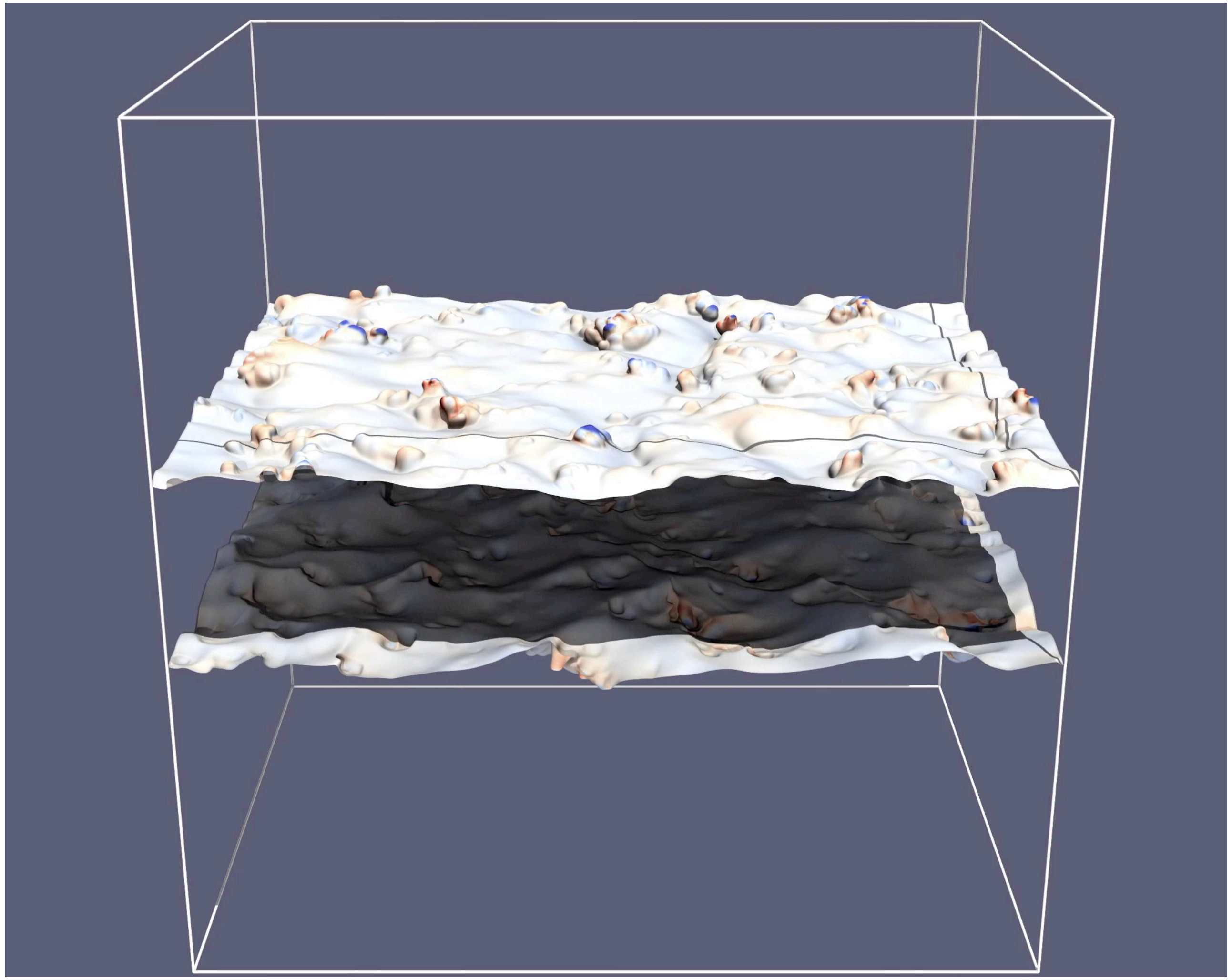


	Low Re low dilution case	Low Re high dilution case	Intermediate Re case	High Da case	High Re case
$\mathrm{Re}_{\mathrm{jet},0}$	4500	4500	6000	6000	10 000
Da_{τ}	0.125	0.150	0.150	0.450	0.150
Z_{st}	0.20	0.45	0.45	0.45	0.45
$n_{\mathrm{gridpoints}}\ [10^9]$	0.4	0.3	0.6	1.6	1.2
DOF $[10^9]$	15	11	23	60	45

DNS of Non-Pemixed Jet Flames

- Iso-surface of the stoichiometric mixture fraction Z_{st}:
  - Optimal mixture between fuel and oxidizer
  - Most probable position of combustion

- Local color indicates the concentration of short lifed species formed in the reaction zone.



DNS of an Engineering Application

Estimate for the computational costs of "realistic" engine conditions on state-of-the-art super computer

Baseline case: Hi *Re* Case $Re \approx 10{,}000, \quad K = 1.5 \cdot 10^7$ CPU-H on JUWELS

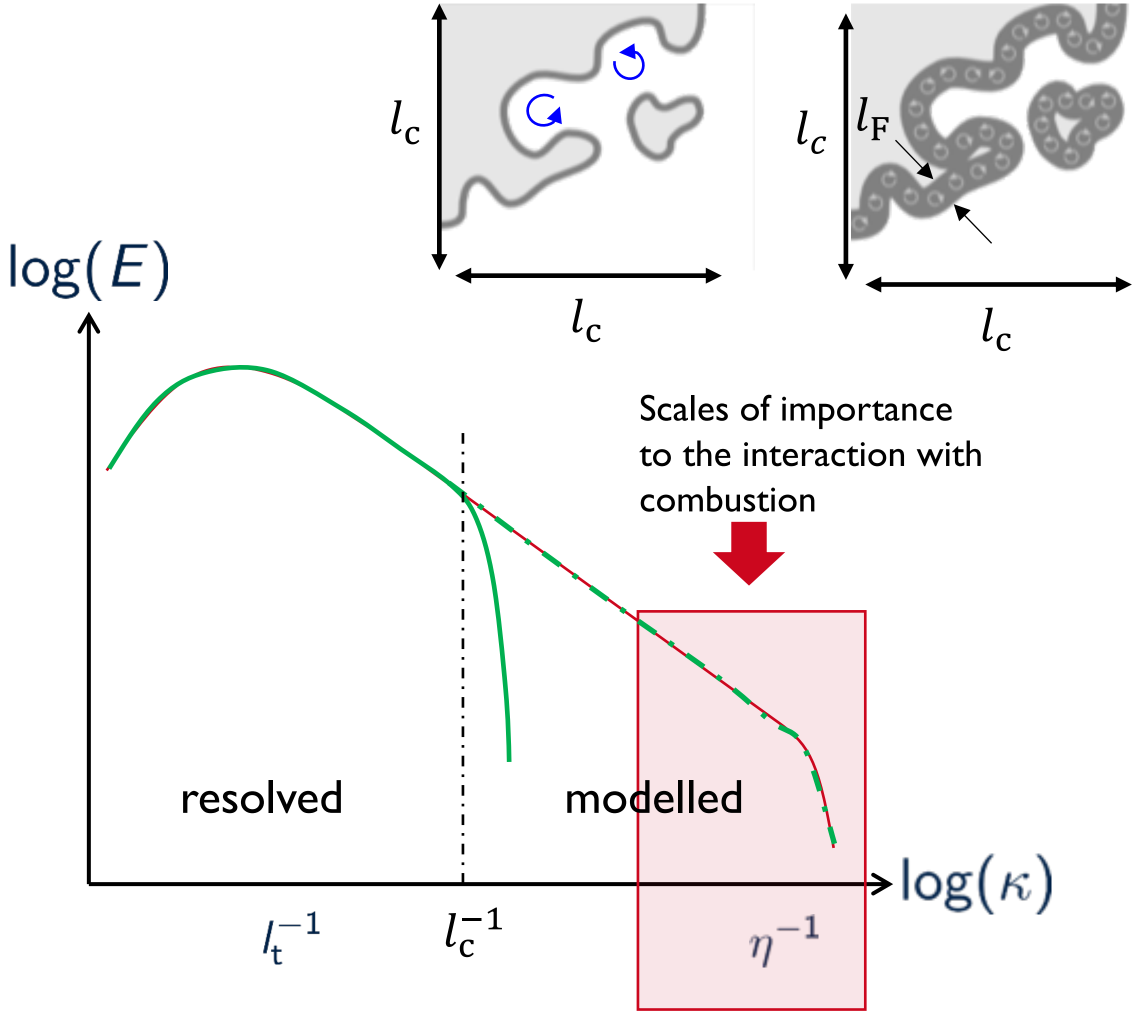
1. Reynolds number in internal combustion engine $Re \approx 100{,}000$
   Cost increase due to scale separation and consequent higher grid resolution:
   $$K' = K \cdot \left[\left(\frac{100{,}000}{10{,}000}\right)^{\frac{3}{4}}\right]^4 = 1.5 \cdot 10^{10} \text{ CPU-H}$$
2. Gasoline fuel with full chemical mechanism instead of Methane with skeletal chemical mechanism
   (3000 Species instead of 30 – 6000 reactions instead of 102):
   $K'' = K' \cdot \frac{3000}{30} = 1.5 \cdot 10^{12}$ CPU-H
3. Non-idealized flow configuration, several iterations ($\mathrm{n} \sim 10^2 - 10^3$) needed for statistical convergence:
   $K_{\text{engine}} = K'' \cdot n = \mathbf{1.5 \cdot 10^{14}}$ **CPU-H**

Not feasible in the near future!

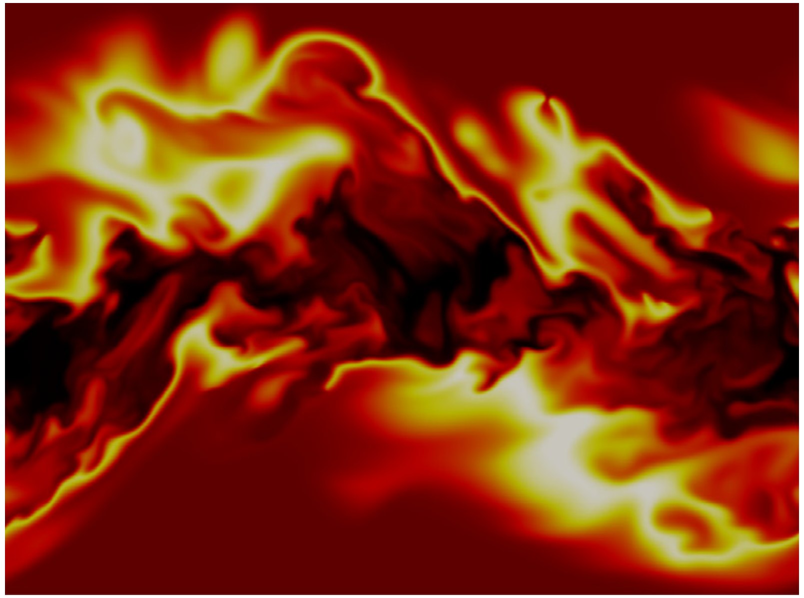
Large Eddy Simulations (LES)

How to simulate turbulent combustion (state-of-the-art)?

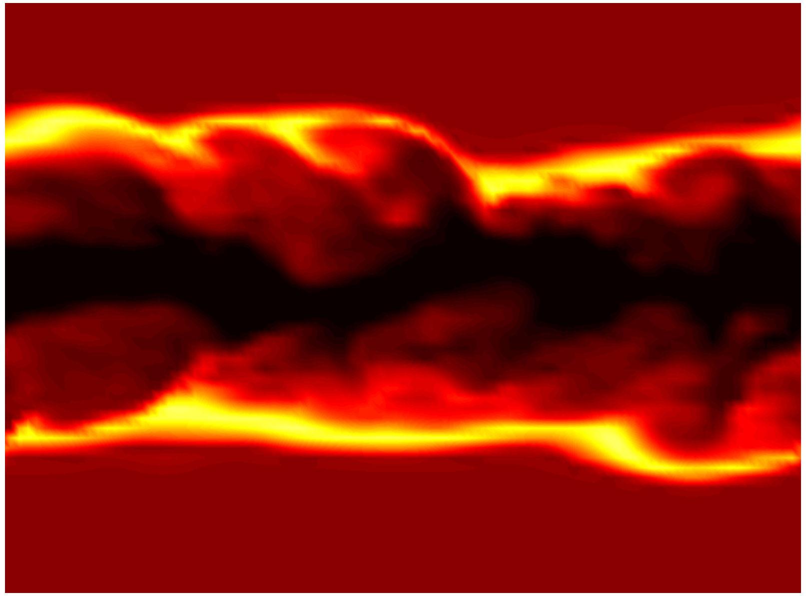
- Simulate only the large, flow-dependent scales "Large Eddies".
- Classical approach: exploit universalities in the small scales in **statistical** models for the "Sub Grid Scales" (SGS)
- SGS models insufficiently capture the highly non-linear interaction between chemistry and fine scale mixing.
- **Solution:**
  Deep Learning - Generate realistic, three-dimensional, and fully resolved turbulent fields



Large Eddy Simulation of Non-Premixed Flame



DNS



LES

Challenges for Artificial Neural Network Training

- Up to 1.2 TB of data generated in each time step
- 10.000 – 12.000 time steps for each DNS case

> More than 600 TB of data from reacting DNS alone

Deep learning at scale

GPU Partitions

CLAIX

- 4 GPU nodes on CLAIX18
  - Platinum 8160 processor
  - 2 Nvidia V100-SXM2 GPUs / node
  - 384GiB memory / node

CLAIX

JURECA

- 1872 compute nodes
  - 75 nodes equipped with 2 Nvidia K80 GPUs / node
  - 2 x 4992 CUDA cores
  - 2 x 24 GiB GDDR5 memory

JURECA

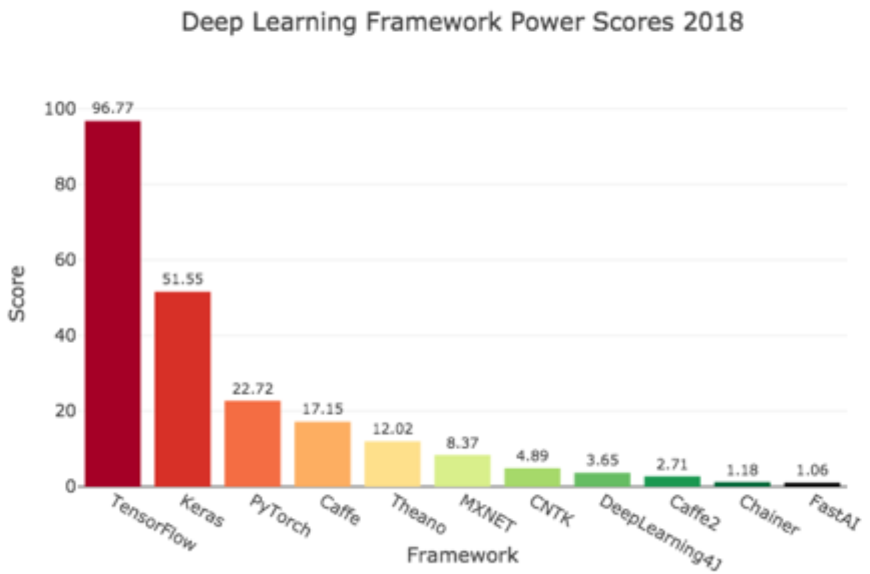
https://doc.itc.rwth-aachen.de/display/CC/Hardware+of+the+RWTH+Compute+Cluster

https://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JURECA/Configuration/Configuration_node.html

Deep learning at scale

Keras

- High-level API for fast neural network prototyping
  - Could be built on different backends, e.g. tensorflow, CNTK or Theano
  - Most frequently used API for various projects
  - Optimal distributed training through first-class support by Horovod
  - More developer friendly than other APIs



Keras

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(MaxPool2D())
model.add(Conv2D(16, (3, 3), activation='relu'))
model.add(MaxPool2D())
model.add(Flatten())
model.add(Dense(10, activation='softmax'))
```

PyTorch

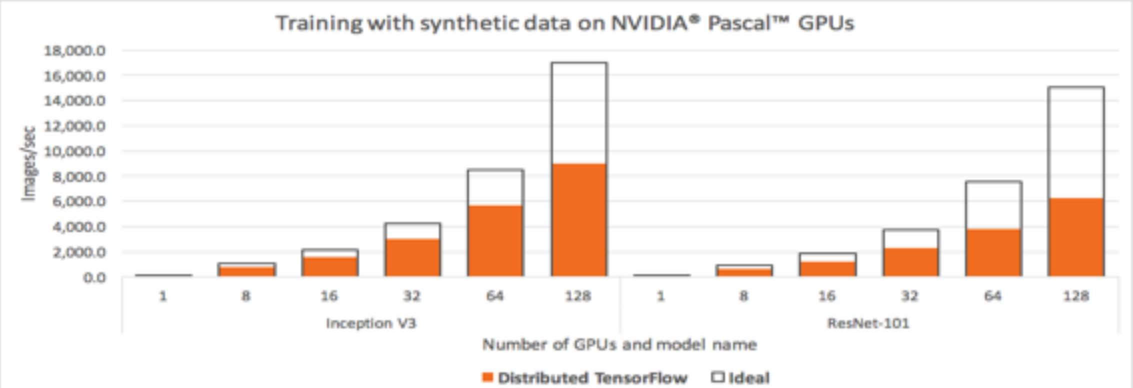
```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(3, 32, 3)
        self.conv2 = nn.Conv2d(32, 16, 3)
        self.fc1 = nn.Linear(16 * 6 * 6, 10)
        self.pool = nn.MaxPool2d(2, 2)
    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        x = x.view(-1, 16 * 6 * 6)
        x = F.log_softmax(self.fc1(x), dim=-1)
        return x
model = Net()
```

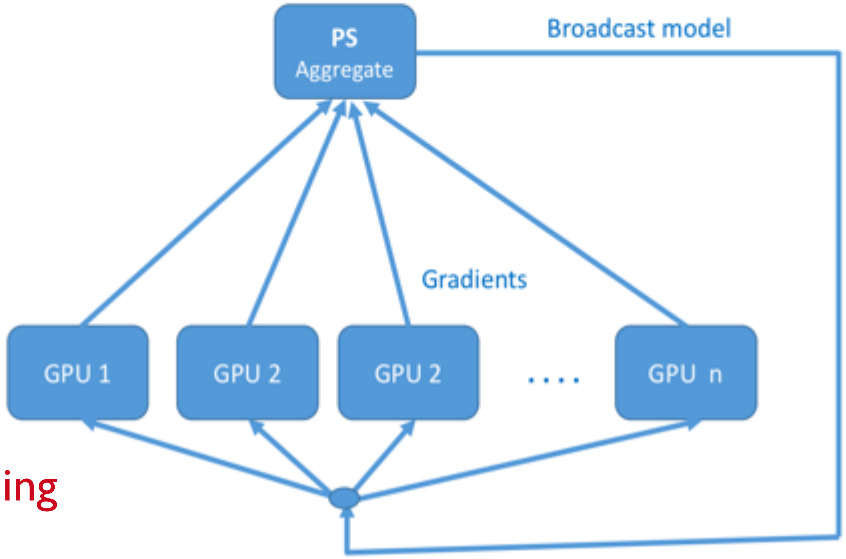

https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a
https://deepsense.ai/keras-or-pytorch/

Deep learning at scale

TensorFlow-GPU

- End-to-end open source platform for building and training machine learning models with GPU support
  - Pros: Low-level tools, flexibility in model features, best library management
  - Cons: Complex implementation, weak benchmarking
- Distributed TensorFlow
  - TensorFlow supports distribution on multiple CPU/GPUs
  - Standard distribution package: *workers, parameter servers, tf.Server(), tf.ClusterSpec(), tf.train_replicas_device_setter()...*
  - These distribution operations introduce hard-to-diagnose bugs→slows training





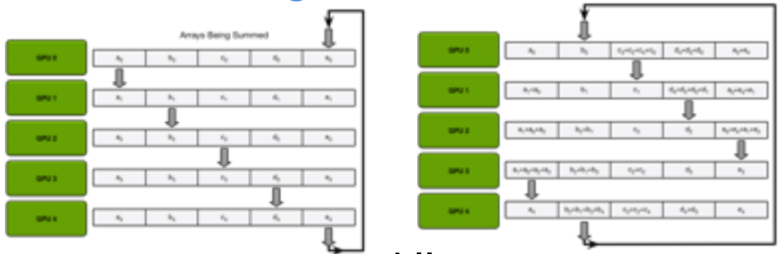
1. Communication cost rapidly grow for increasing GPUs
2. Server must wait for till all GPUs finish → ideling

https://eng.uber.com/horovod/
https://towardsdatascience.com/distributed-tensorflow-using-horovod-6d572f8790c4

Deep learning at scale

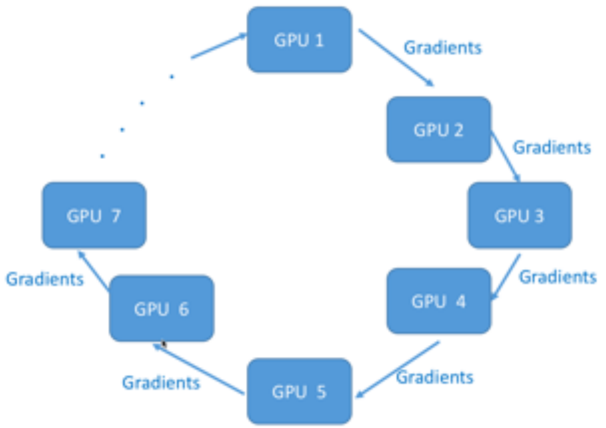
Horovod

- Package to speed-up distributed deep learnings
  - Uses Ring-Allreduce model := All-scatter+All-gather

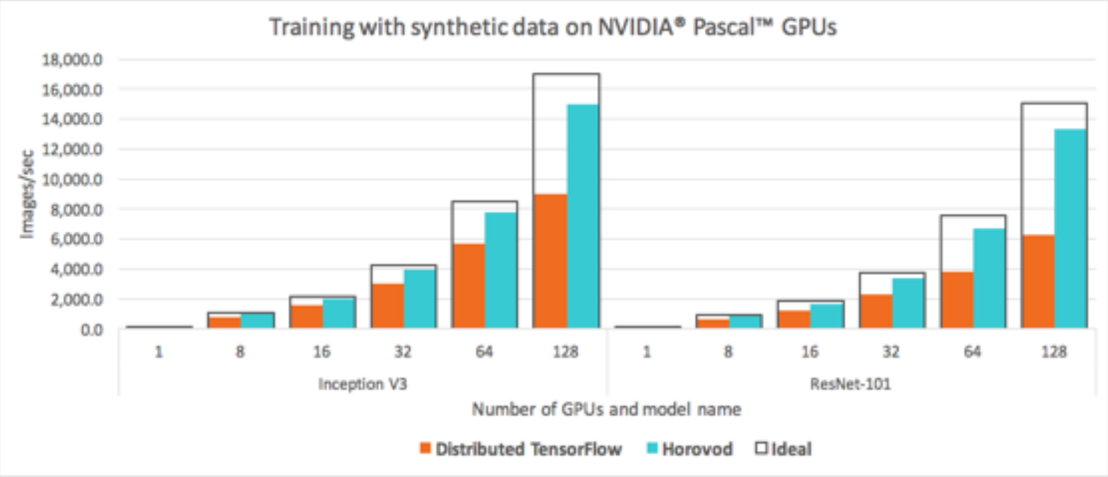


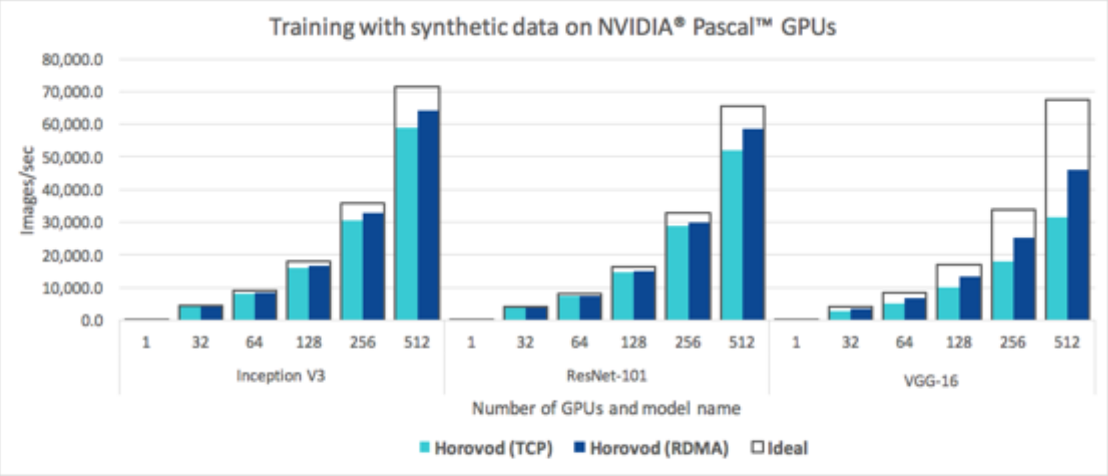
All-scatter

All-gather



  - Improves the scaling efficiency from 50% to 90% for both Inception V3 and ResNet-101





https://eng.uber.com/horovod/
https://towardsdatascience.com/distributed-tensorflow-using-horovod-6d572f8790c4
https://andrew.gibiansky..com

Deep learning at scale

Challenges

- Communication
  - Bottleneck on first rank
  - Communication tree with recursive broadcast
- I/O
  - GPFS speed limited
  - Distributed data staging
  - Point-to-point MPI

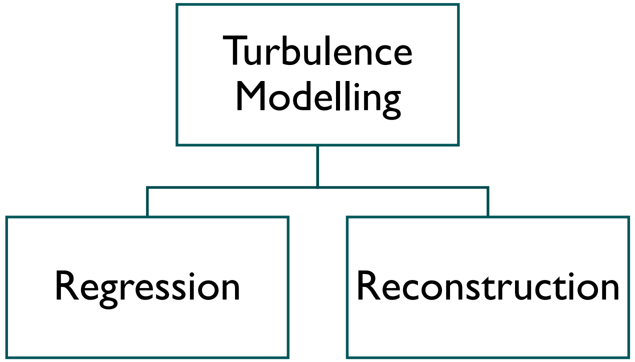
Nodes	Ideal [TFLOPS]	Before [TFLOPS]	After [TFLOPS]
1	30	30	30
10	300	291	290
50	1500	702	1480
75	2250	1003	2023

Modelling

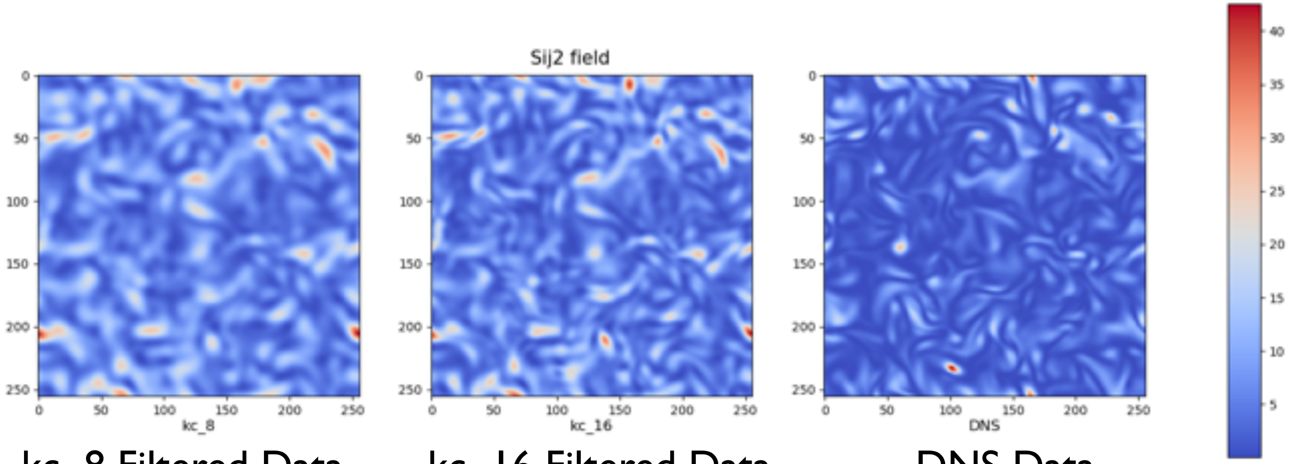
2 Data-Driven Turbulence Modelling Approaches

1) Regression: Fully Connected Artificial Neural Network to predict certain turbulence parameters using other parameters

2) Reconstruction: Artificial Neural Network to reconstruct fully resolved, DNS turbulence fields from low resolution data.



DNS data



kc_8 Filtered Data | kc_16 Filtered Data | DNS Data

GAN (Generative adversarial networks)

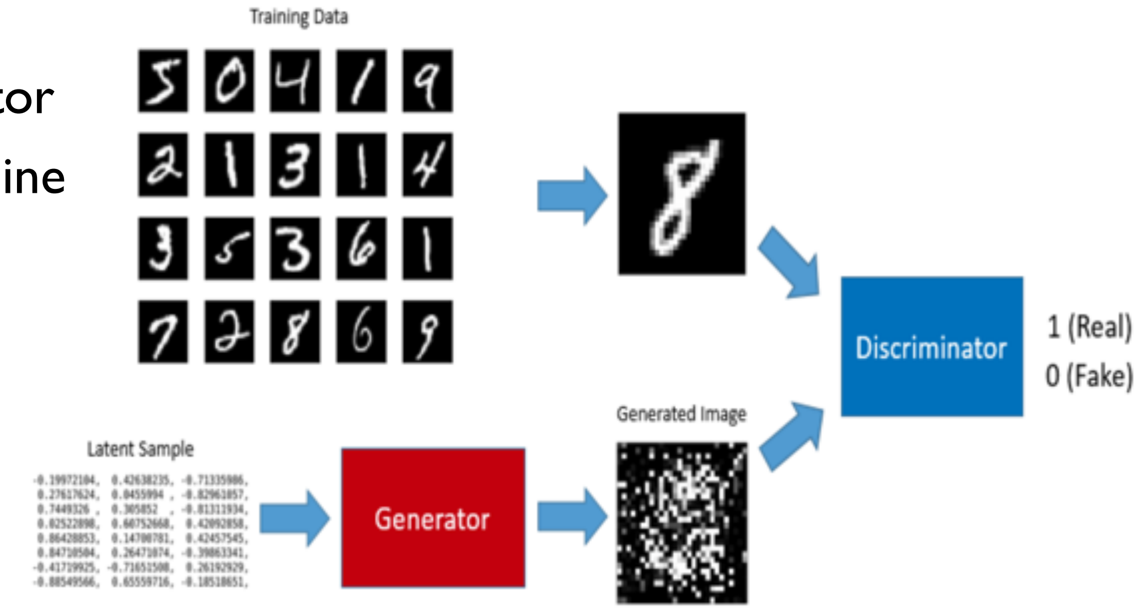
- What is GAN:

  GAN includes a generator and a discriminator

  - Generator: captures data distribution, tries to produce "real" samples that would hopefully fool the discriminator
  - Discriminator: judges whether the input sample is genuine or "faked" produced by the generator

- Why GAN:

  - It is generative
  - Important for unsupervised learning
  - GAN maps one probability distribution to another

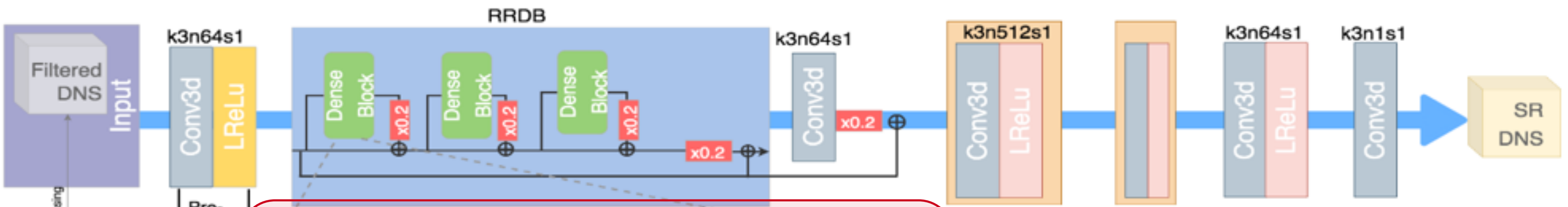


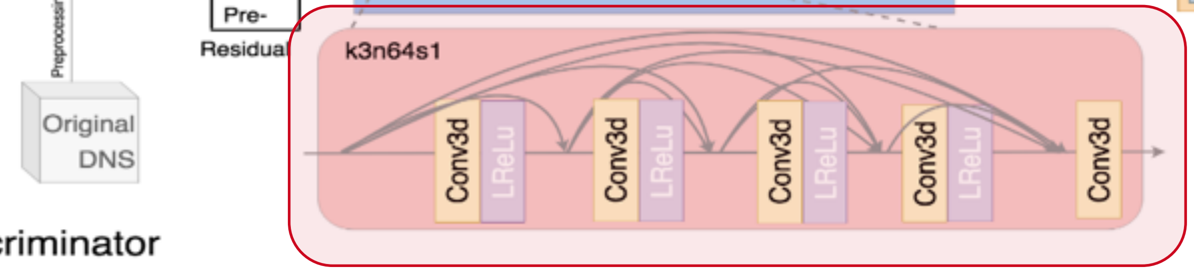
Structure demonstration of GAN

Reconstruction

- Network structure

Generator





The convolution allows for the consideration of the multi-scale nature of turbulence

Discriminator

PIESRGAN (physics-informed enhanced super-resolution GAN)

- Derived from the 2D image ESRGAN framework

  - Uses convolutional layers for feature (turbulence eddies) extraction

  - We use the DNS data, and its filtered data as inputs. The data includes e.g.:

velocity components	passive scalar	velocity gradients	Reynolds number	filter width	dissipation rate

  - Applies a residual-in-residual dense block (RRDB) in the generator model, which greatly increases the model complexity through jump communications

  - A noval concept for the cost function: physical-based loss

    1. For passive scalar: MSE-loss of the gradient field

    $$l_{grad} = \frac{1}{N_{sample}}(\nabla_{x_i}\phi_{true} - \nabla_{x_i}\phi_{pred})^2$$

    2. For velocity reconstruction: continuity loss

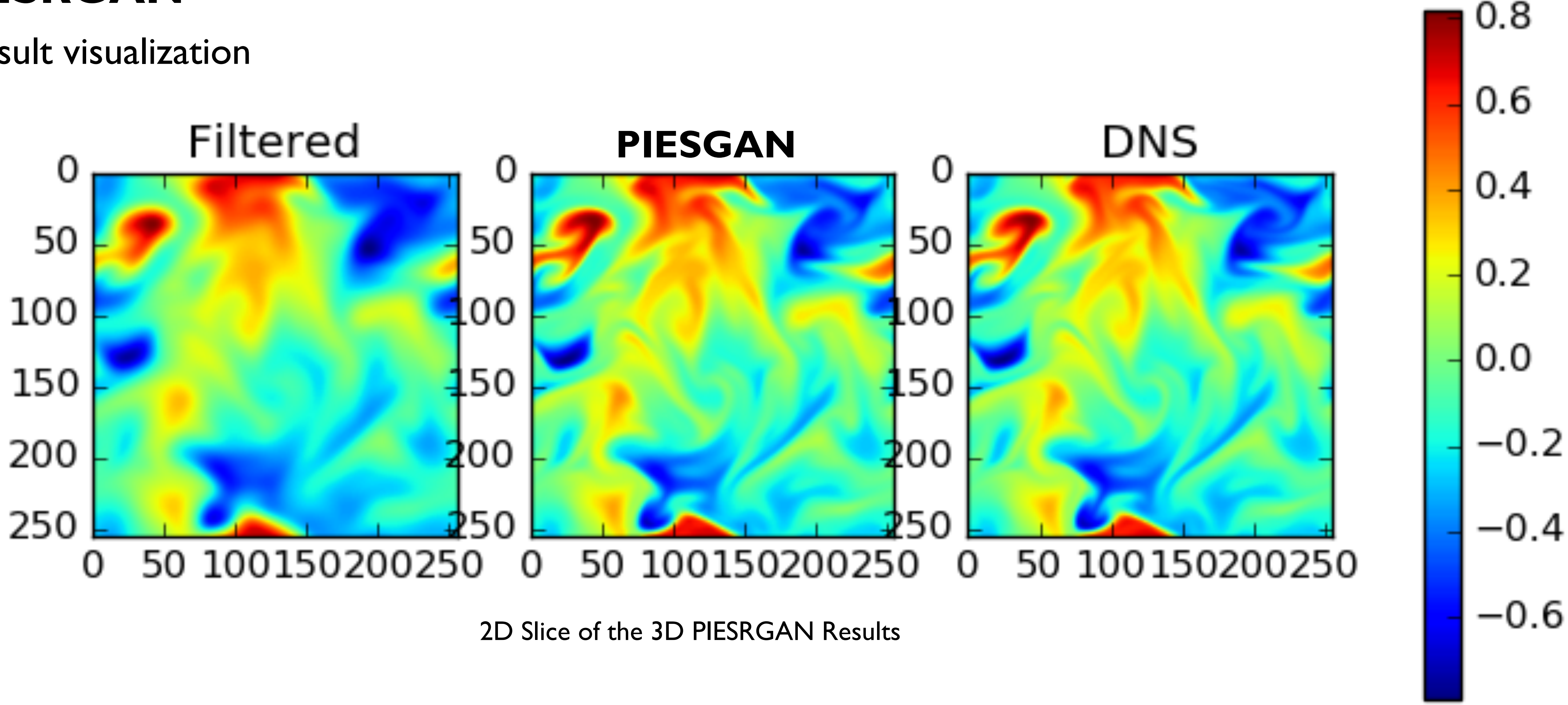
    $$l_{conti} = \frac{1}{N_{sample}}(\nabla \cdot \phi_{pred})^2$$

2D ESRGAN validation
(L)orignal LR (M) bilinear (R) ESRGAN

Reconstruction

PIESRGAN

- Result visualization



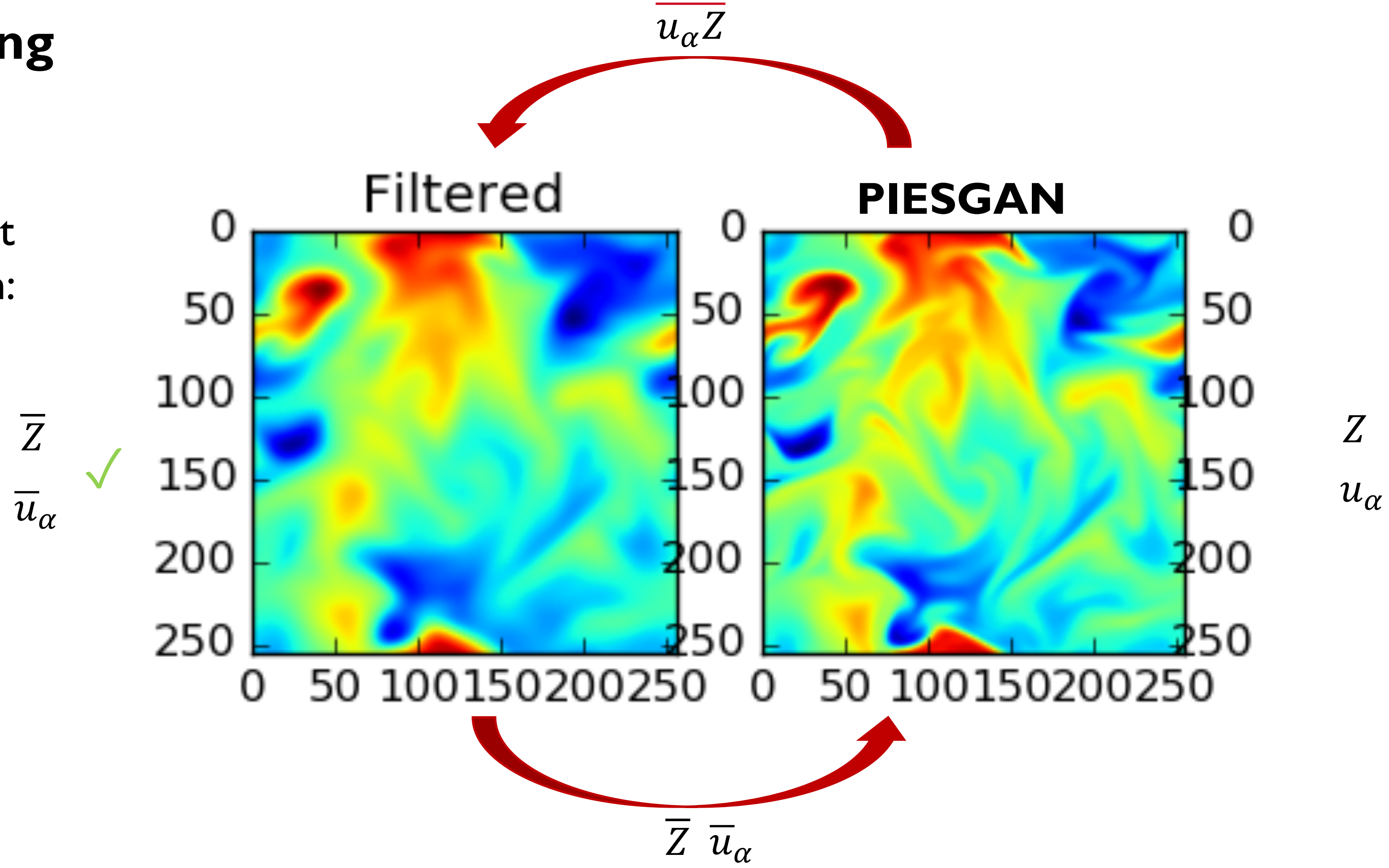
2D Slice of the 3D PIESRGAN Results

A posteriori testing

Sub-filter modeling

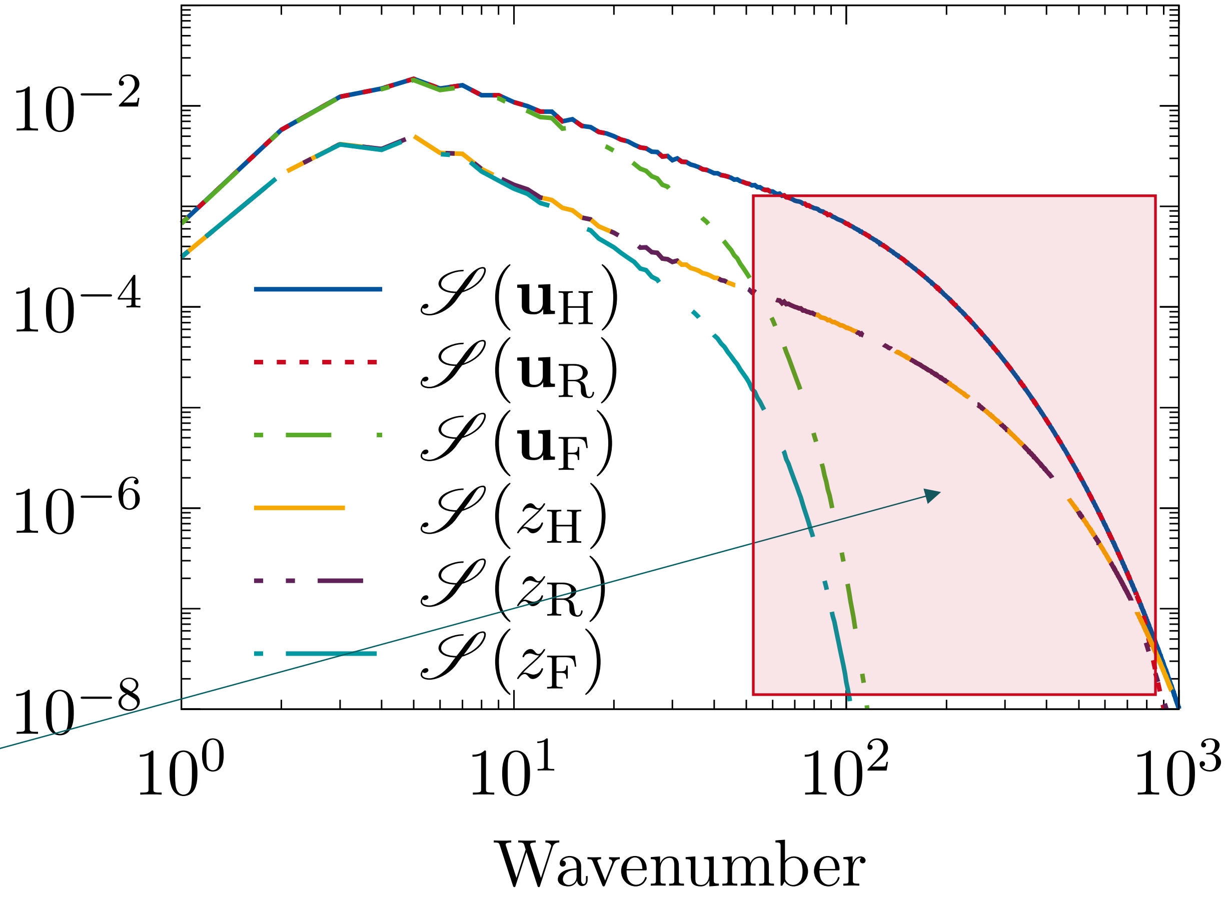
- Unclosed terms in filtered equations, for example SGS transport of the mixture fraction:

$$\frac{\overline{\partial u_\alpha Z}}{\partial x_\beta} = ?$$



Validation of the results: energy spectrum

- Energy spectrum provides scale-dependent validation of the accuracy of the PIESRGAN
- Filtered (LES) solution lacks information at the small scales, which is provided BY the PIESRGAN
- PIESRGAN is able to predict small-scale turbulence and close the LES equations

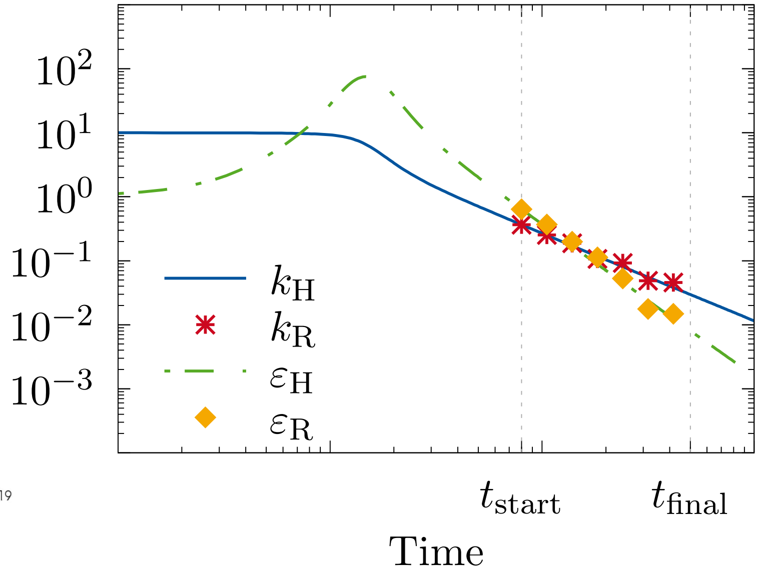


Information about SGS is provided by PIESRGAN

A posteriori testing

Turbulence

- Application of PIESGAN-SGS model for LES of decaying turbulence
- Good agreement of statistics
- Questions:
  - Using this model for higher Reynolds number?
  - Performance in multi-physics cases?

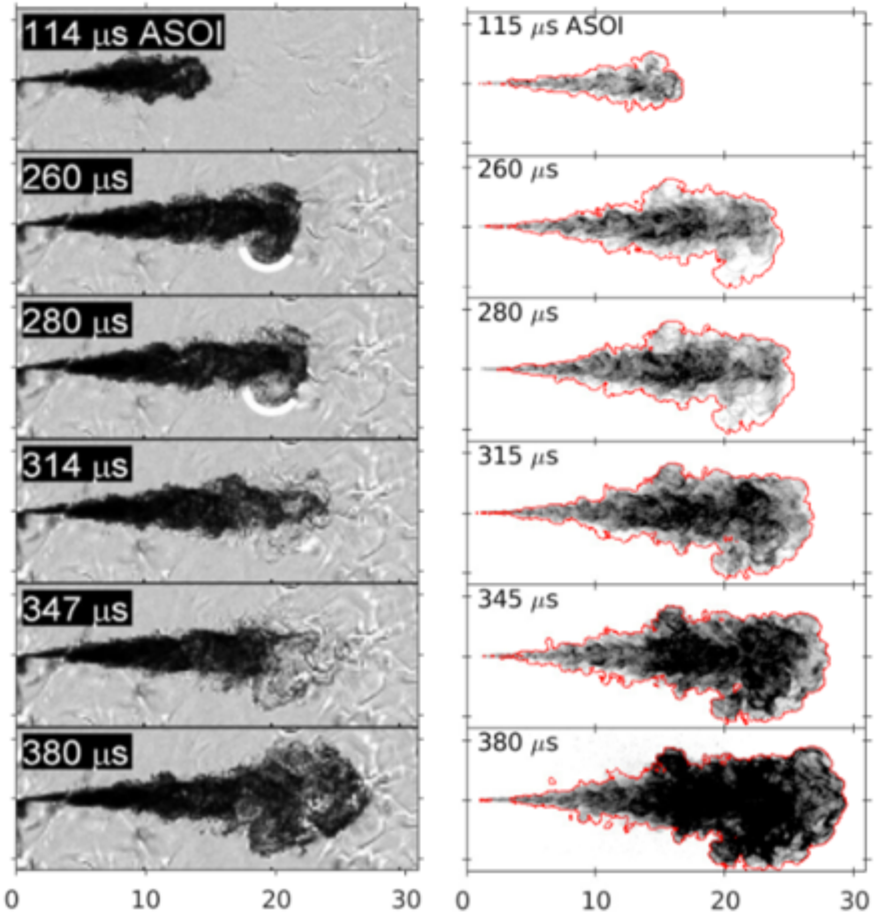


DNS: 4096^3 grid points
LES: 64^3 grid points

A posteriori testing

Spray case

- Application of PIESRGAN-SGS model for LES of decaying turbulence
- Application of 5-Layer Dense ANN for chemistry
- Reduction of computing time to 57%
- Ignition delay times: 0.435 ms (SGS) vs. 0.421 ms (PIESRGAN-SGS)
- Flame lift-off: 13.4 mm (SGS) vs. 13.1 mm (PIESRGAN-SGS)



Conclusions

- Motivation and introduction to turbulent combustion
- Generation of DNS Combustion data explained
- Deep learning at scale is possible if bottlenecks are removed
- PIESRGAN as network for modeling introduced
- A posteriori testing results show good accuracy

Thank you for your attention

Dominik Denker

Institute for Combustion Technology
RWTH Aachen University

http://www.itv.rwth-aachen.de