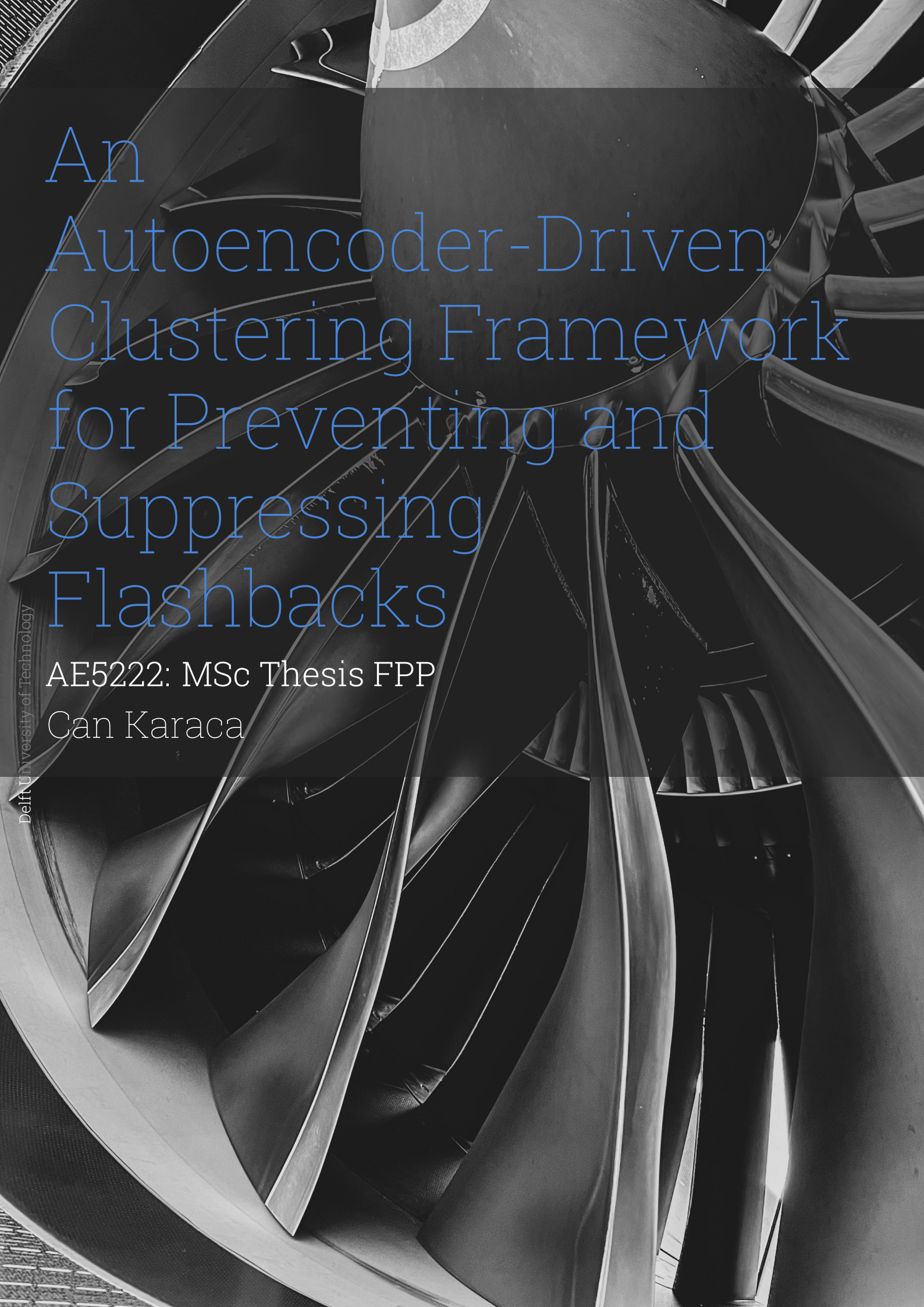


An Autoencoder-Driven Clustering Framework for Preventing and Suppressing Flashbacks

AE5222: MSc Thesis FPP

Can Karaca

Delft University of Technology

An Autoencoder-Driven Clustering Framework for Preventing and Suppressing Flashbacks

by

Can Karaca

Student Number: 5211123

Thursday 23rd January, 2025 - Tuesday 16th September, 2025

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

Preface

Firstly, I owe my deepest gratitude to my supervisor, Dr. Ivan Langella, whose guidance, insightful feedback, and constant encouragement have been invaluable throughout this project. His mentorship not only helped me refine my work but also opened pathways into fascinating and promising areas of research, further strengthening the passion that initially drew me to this topic. I am equally thankful to Mihnea Floris, whose consistent support and perspective were instrumental in shaping the direction and progress of this thesis.

I would also like to extend my heartfelt thanks to my friends who have stood by me during these five years at Delft. Whether through long evenings playing Smash or games of volleyball, their friendship has kept me grounded and reminded me of the importance of balance and joy amid the challenges of demanding degrees. Their presence has been a large part of my journey, and I would not be where I am today without them.

Finally, I am profoundly grateful to my family, whose unwavering belief in me and continuous support made this degree possible. Their encouragement has been a source of strength through every step of this academic path.

As this chapter of my life in Delft comes to a close, I look back with gratitude and fond memories, and I look forward to carrying these experiences into the next stage of my journey.

Abstract

This research addresses the challenge of thermoacoustic instabilities and flashback in hydrogen combustion under low-emission conditions. The study focuses on predicting the onset of flashback in a simplified model of Ansaldo Energia's GT36 reheat combustor, simulated at high pressure (20 bar) using Large Eddy Simulation (LES). Under lean, premixed conditions, the LES reveals unsteady flame dynamics driven by strong pressure oscillations, leading to repeated autoignition events in the mixing duct.

To this end, a data-driven framework was developed that combines LES-derived time series with dimensionality reduction via autoencoders and state identification through clustering. Instead of relying on idealized in-flame probes, signals were extracted at the combustor wall, representing a step toward practical sensor placement. An analysis of spatial correlations was first carried out to identify suitable wall locations. Fourteen thermodynamic, velocity, and species mass fraction signals were then monitored across multiple flashback events. Autoencoders with bottlenecks of two, three, and four latent variables were trained to compress these correlated signals into compact trajectories. The three-latent representation emerged as optimal: it preserved the cyclic structure of stable operation while isolating transition sharpness and mid-frequency modes.

Clustering applied to this three-latent space, using a modularity-based graph clustering algorithm, proved highly effective in predicting flashback. In this approach, the latent trajectories are tessellated into discrete hypercubes, converted into a graph representation, and then partitioned by maximizing modularity to reveal distinct dynamical states. Precursors were consistently identified with a maximum lead time of ~ 42 μs, sufficient for active control, with 1 very small false positive during the rapid flashback regime. Crucially, in the most severe case, a flashback event was successfully predicted and suppressed, establishing the practical applicability of the framework for real-time instability mitigation.

Robustness analyses confirm the generality of these findings. Architectures with two latents failed to capture transition dynamics, while four latents provided only incremental improvements, demonstrating diminishing returns. Extreme-value robustness checks showed that rare outliers do not compromise detection. Chronological validation confirmed generalization to unseen segments, and testing at a second wall monitoring location revealed consistent performance for thermo-chemical variables, with degradation confined to highly noisy channels such as pressure and transverse velocities. Together, these results show that the clustering framework not only captures fundamental instability features but also advances toward real-world feasibility by operating on wall-based measurements, bringing predictive flashback control closer to deployment in practical gas turbine environments.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
AE	Autoencoder
ADAM	Adaptive Moment Estimation (optimizer)
AI	Artificial Intelligence
AMR	Automatic Mesh Refinement
BAT	Best Available Technique
BAT-AEEL	Best Available Technique – Associated Energy Efficiency Level
BCE	Binary Cross-Entropy
CFD	Computational Fluid Dynamics
CIVB	Combustion-Induced Vortex Breakdown
CNN	Convolutional Neural Network
CORSIA	Carbon Offsetting and Reduction Scheme for International Aviation
CPSC	Constant Pressure Sequential Combustion
CDZ	Central Developing Zone
DLN	Dry Low NO_x (burner)
DNS	Direct Numerical Simulation
FN	False Negative
FP	False Positive
FVM	Finite Volume Method
GHG	Greenhouse Gas
HRSG	Heat Recovery Steam Generator
IEA	International Energy Agency
IPCC	Intergovernmental Panel on Climate Change
LES	Large Eddy Simulation
LHV	Lower Heating Value
LNB	Low NO_x Burner
LSTM	Long Short-Term Memory (neural network)
MAE	Mean Absolute Error
MET	Mean Exit Temperature
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NSCBC	Navier–Stokes Characteristic Boundary Condition
ORZ	Outer Recirculation Zone
PCA	Principal Component Analysis
PPMVD	Parts per Million per Volume, Dry
RANS	Reynolds-Averaged Navier–Stokes

Abbreviation	Definition
ReLU	Rectified Linear Unit (activation function)
RFI	Radiative Forcing Index
RNN	Recurrent Neural Network
RMS	Root Mean Square
ROM	Reduced Order Model
RQA	Recurrence Quantification Analysis
SCR	Selective Catalytic Reduction
SGD	Stochastic Gradient Descent
SGT	Siemens Gas Turbine (series)
SMD	Sauter Mean Diameter
SOR	Successive Over-Relaxation
TFM	Thickened Flame Model
UNFCCC	United Nations Framework Convention on Climate Change
VAE	Variational Autoencoder
VAE-POD	Variational Autoencoder – Proper Orthogonal Decomposition
WLN	Wet Low NO_x (burner)

Symbols

Symbol	Definition
α	Thermal diffusivity of the mixture; also used as filtering parameter
$\alpha^{[k]}$	Coefficient in Taylor expansion approximation
α_1	Cold/hot state parameter for flame sensor
β	Cone angle (spray geometry); also used as modeling parameter for flame sensor thickness
β_{Colin}	Colin model parameter for efficiency formulation
β_l	Eigenvalue corresponding to eigenvector v_l
δ_{ij}	Kronecker delta
δ_l	Laminar flame thickness
Δ	LES filter width (cell size scale)
Δx	Local grid spacing
Δx_{base}	Baseline mesh size
Δx_{new}	Refined mesh size
ϵ	(i) Small constant in Adam optimizer, (ii) Distance to cluster centroid (context dependent)
Γ_Δ	Efficiency function for turbulence scales smaller than Δ
η	Learning rate
μ	Dynamic viscosity
ν	Kinematic viscosity (if used, check consistency)
$\Omega_{\text{sens},0}$	Maximum reaction rate of the sensor at equivalence ratio ϕ
ρ	Density

Symbol	Definition
ρ_L	Liquid density
σ	Surface tension
σ_k	Reciprocal subgrid-scale kinetic energy constant
τ_0	Local relaxation time
τ_1	Relaxation time scaled by α, $\tau_1 = \alpha\tau_c$
τ_c	Characteristic flame time, $\tau_c = \delta_l / s_L$
ϕ	Total scalar field; also used as neural network activation function
ϕ'	Subgrid scalar field
$\bar{\phi}$	Resolved scalar field
ψ	Passive indicator function
ω	Reaction rate
$\bar{\omega}_k$	Filtered chemical reaction rate of species k
$\tilde{\omega}_\psi$	Relaxation source term linked to ψ
b_j	Bias term associated with neuron j
c_m	Model constant in Colin's model
c_p	Specific heat at constant pressure
d_0	Injection/nozzle diameter
e	Specific energy
h_m	Species-specific enthalpy of species m
k	Thermal conductivity
k_i	Degree of vertex i in a graph
$k_r(x,t)$	Resolved-scale turbulent kinetic energy
m	Total number of edges (graph context); also index for species
$m(B_i)$	Number of phase space points in hypercube B_i
n	(i) Number of inputs/features, (ii) Number of vertices in a graph (context dependent)
r	Radius (droplet or geometry)
s	Mesh refinement level
s_i	Community membership of vertex i, with $s_i \in \{+1, -1\}$
s_L	Laminar flame speed
u, v, w	Velocity components (streamwise, transverse, spanwise)
u_t	Turbulent velocity magnitude
$\tilde{u}, \tilde{v}, \tilde{w}$	Time-averaged RMS velocity perturbations
u'_Δ	Local turbulent velocity fluctuation at scale Δ
w_{ij}	Neural network weight connecting neuron i to neuron j
x_i	Output of neuron i from previous layer
y_j	Output of neuron j in neural network
Y_i	Mass fraction of species i
Y_m	Mass fraction of species m
X_m	Mole fraction of species m
A_{ij}	Entry of adjacency matrix
B_i, B_j	Hypercubes in discretized phase space

Symbol	Definition
$\mathbf{B}$	Modularity matrix
B^T	Transpose of modularity matrix
C_ϵ	Subgrid-scale dissipation constant
C_K	Viscosity constant (subgrid-scale model)
E	Efficiency factor for flame–turbulence interaction
E_A	Activation energy
F	LES filter operator (context: filtering)
F	Local flame sensor factor (context: combustion modeling)
$F_{\max}$	Maximum flame sensor value
$\mathcal{F}^1$	Temporal forward operator
$\mathcal{F}^{-1}$	Temporal backstep operator
J_j^k	Filtered laminar diffusion flux of species k
J_j^h	Filtered enthalpy diffusion flux
K	Turbulent kinetic energy (resolved scales)
K_t	Turbulent (effective) thermal conductivity
L	Latent heat of evaporation
$M(x,t)$	Turbulence resolution metric (Pope's criterion)
N	Total number of hypercubes
$\mathcal{P}$	Transition probability matrix
P	Pressure
P_{ij}	Transition probability from hypercube B_i to B_j
Pr_t	Turbulent Prandtl number
Q	Graph modularity metric
R	Universal gas constant
R_d	Droplet radius
Re_L	Reynolds number based on length L
Re_Δ	Subgrid-scale Reynolds number
Re_t	Turbulent Reynolds number
S	Source term (generic); also local flame sensor
S_m	Source term for species m (e.g., evaporation)
T_b	Burnt gas temperature
T_u	Unburnt gas temperature
$\bar{T}$	Mean temperature between unburnt and burnt gases
T_s	Switching temperature for α_1
V	Computational cell volume
V_{inj}	Injection velocity
V_{response}	Response velocity
V_n	Normal velocity component to a surface
$\tilde{Q}$	Favre-filtered variable
$\bar{Q}$	Favre-averaged variable
$\bar{\rho}$	Favre-averaged density
Ξ_Δ	Efficiency function for turbulence straining at the flame front
ψ	Passive indicator function
$\mathbf{s}$	Community assignment vector

Symbol	Definition
v_l	Eigenvector of $(B + B^T)$
a_l	Coefficient of eigenvector v_l
$(v_l^T \mathbf{s})^2$	Projection of community assignment vector onto eigenvector v_l

1

Introduction

1.1. Hydrogen as a Fuel

The decarbonisation of energy systems is among the most pressing challenges of the twenty–first century. In particular, the global effort to mitigate climate change requires a rapid reduction of greenhouse gas (GHG) emissions from the power generation and industrial sectors. Gas turbines play a central role in this transition: they offer high power density, rapid load–following capability, and compatibility with existing infrastructure. Their flexibility makes them indispensable not only in conventional power plants but also as backup for renewable sources with fluctuating output.

Hydrogen has emerged as a leading candidate to replace or complement natural gas in turbine applications. It is a carbon–free fuel that, when produced from renewable electricity via electrolysis, contributes directly to the decarbonisation of energy supply chains. However, the use of hydrogen in gas turbines also introduces substantial technical challenges. Among these are thermoacoustic instabilities, flashback, and increased susceptibility to autoignition due to hydrogen's wide flammability limits, high flame speed, and low ignition energy. These instabilities can result in significant efficiency losses, accelerated component wear, or catastrophic failure.

Flashback, in particular, is a critical hazard. It occurs when the flame propagates upstream into the premixing section of the combustor, leading to potential hardware damage and operational shutdown. While conventional burners are designed with flame stabilisation mechanisms, hydrogen's reactivity and diffusivity increase the risk of flashback under lean premixed conditions that are otherwise desirable for low NOx emissions. Detecting flashback precursors early and suppressing them before damage occurs is therefore a key enabler for hydrogen–fired turbine technology.

1.2. Methods

Traditional approaches to modelling and predicting flashback rely on physics–based tools such as Reynolds–Averaged Navier–Stokes (RANS) and Large Eddy Simulation (LES). LES, in particular, has proven effective at resolving the unsteady flame dynamics that precede instabilities, capturing both cycle–to–cycle variability and transient excursions. However, LES produces massive, high–dimensional datasets: hundreds of thousands of grid points and dozens of variables, sampled at microsecond resolution. Manually extracting reliable precursor signatures from such data is infeasible.

In parallel, recent advances in machine learning (ML) have opened new avenues for identifying precur-

sors in complex, multiscale systems. Autoencoders, and more generally deep learning architectures, are able to reduce high–dimensional correlated data into low–dimensional latent spaces that preserve essential dynamics, called feature extraction. These latent spaces offer interpretable coordinates in which cycles, transitions, and stochastic noise can be disentangled. When coupled with clustering algorithms, they provide a natural way to identify state changes and isolate early precursors of instability.

This thesis builds on these opportunities by developing a data–driven framework for flashback prediction. Starting from high–fidelity LES data of a hydrogen–fired reheat combustor, a set of thermodynamic, velocity, and species mass fraction time series are extracted near the flame front. These signals are then compressed using autoencoders with bottlenecks of two to four latent variables, producing compact representations that can be systematically compared. The latent trajectories are subsequently subjected to clustering analysis to detect precursor states that precede flashback events. The ultimate objective is not merely dimensionality reduction but actionable prediction: a reliable signal that flashback is imminent, available early enough to enable countermeasures such as water injection.

1.3. Challenges

While the outlook for hydrogen as a low–carbon energy carrier is promising, several practical challenges remain. First, hydrogen combustion poses intrinsic physical difficulties. Its high reactivity, wide flammability limits, and low ignition energy make it prone to thermoacoustic instabilities and flashback, phenomena that are harder to predict and control than in conventional hydrocarbon systems. Beyond the technical aspects, public perception and safety concerns also complicate adoption: hydrogen is often regarded as hazardous due to its diffusivity and history of high–profile accidents, which places a premium on reliable monitoring and early–warning systems.

Second, the monitoring strategy itself introduces substantial difficulty. Most prior research has relied on flame–front measurements, where the signals are rich in information and strongly correlated with combustion dynamics. In contrast, industrially realistic sensors must be placed at the combustor wall, where the signals are noisier, less direct, and more easily distorted by acoustic reflections or boundary effects. Extracting precursor information from such wall data is significantly more challenging, yet essential for practical applicability.

Finally, predictive frameworks must ultimately be judged on operational metrics: the ability to provide sufficient precursor time for control action, while minimizing false positives that could lead to unnecessary interventions and false negatives that risk catastrophic events. Obtaining a balance between sensitivity, robustness, and practicality is a large challenge in moving towards deployable hydrogen combustion monitoring.

1.4. Objectives

The main objective of this thesis is to design and validate a data–driven framework for flashback precursor detection in hydrogen combustion, using LES data and machine learning methods. The work pursues the following goals:

- To extract and preprocess time–series signals of thermodynamic, velocity, and species variables from LES simulations of a reheat combustor under flashback–prone conditions, at practical locations such as the wall
- To design, train, and evaluate autoencoders with two to four latent variables, systematically assessing their reconstruction fidelity, error distributions, and latent trajectories.
- To identify the latent dimensionality that best balances compression and expressivity, with particular emphasis on capturing transition dynamics preceding flashback.

- To apply clustering in the optimal latent space to detect precursor states, and to quantify prediction performance in terms of average precursor time, maximum precursor time, and rates of false positives/negatives.
- To demonstrate the ability of the method to suppress flashback events by enabling countermeasures (e.g. water injection) within the available precursor time.
- To assess robustness by testing on unseen locations, extreme values, and reduced feature sets, ensuring that the framework generalises beyond the training configuration.

In doing so, the thesis aims to establish a reproducible pathway from high–fidelity simulation data to practical instability prediction tools. By quantifying precursor times, the work contributes to the feasibility of online monitoring and control of hydrogen gas turbines, thereby supporting the broader transition toward low–emission energy systems.

2

Emission Technology

This chapter provides an overview of emission technologies relevant to power generation and aerospace applications. It begins by outlining current greenhouse gas (GHG) emissions and the global efforts to mitigate them. Following this, the chapter explores the energy sector's impact on emissions, highlighting the continued reliance on fossil fuels and the shift towards renewable alternatives. The role of gas turbines in both energy and aerospace industries is then examined, focusing on major manufacturers and their advancements in low-emission combustion technologies. Finally, special attention is given to Ansaldo Energia's GT36 gas turbine, which incorporates sequential combustion to enhance efficiency and reduce emissions.

2.1. Current Emissions

One of the largest discussion topics in today's society is where global emissions are headed. Using methods to generate power in current society has been useful for development. Still, it has caused the release of tonnes of GHG that will continue to have negative effects for years. Jones et al. [67] performed a study on the emissions of global GHG in the past years. This study is graphed in Figure 2.1.

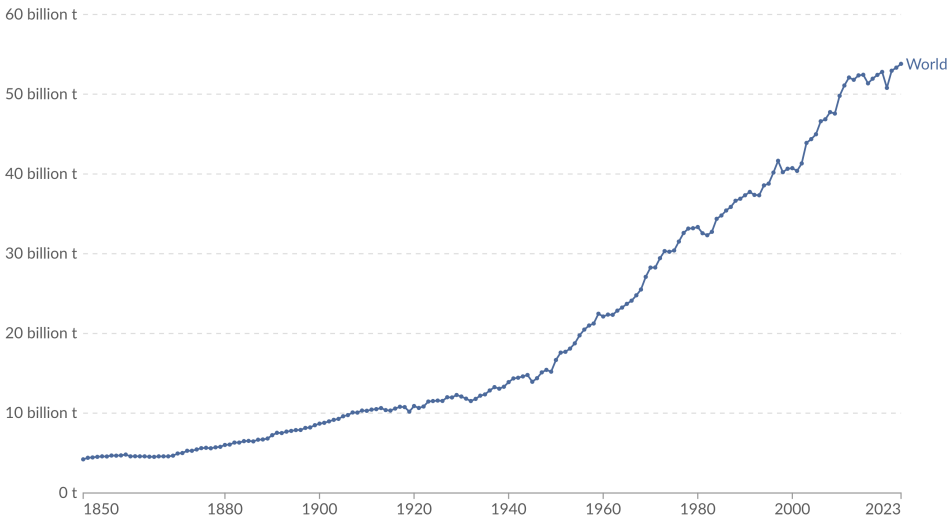


Figure 2.1: Global GHG Emissions [67]

Over the past several decades, international efforts to reduce GHG emissions have evolved through a series of significant agreements and initiatives. The United Nations Framework Convention on Climate Change (UNFCCC), established in 1994, marked the first major international treaty aimed at addressing climate change by stabilizing GHG concentrations to prevent dangerous anthropogenic interference with the climate system [100]. Building upon this foundation, the Kyoto Protocol was adopted in 1997, setting legally binding emission reduction targets for developed countries [119]. In 2015, the Paris Agreement represented a landmark accord, with nearly all countries committing to limit global warming to below 1.5° C though progress has been mixed with global CO_2 emissions continuing to rise, underscoring the need for more effective implementation [82].

To address the root causes of the issue, it is essential to first identify the sectors that contribute most to emissions. The European Parliament [91] collects data on GHG emissions within the EU, especially regarding breakdown per sector. These findings for 2022 are presented in Figure 2.2.

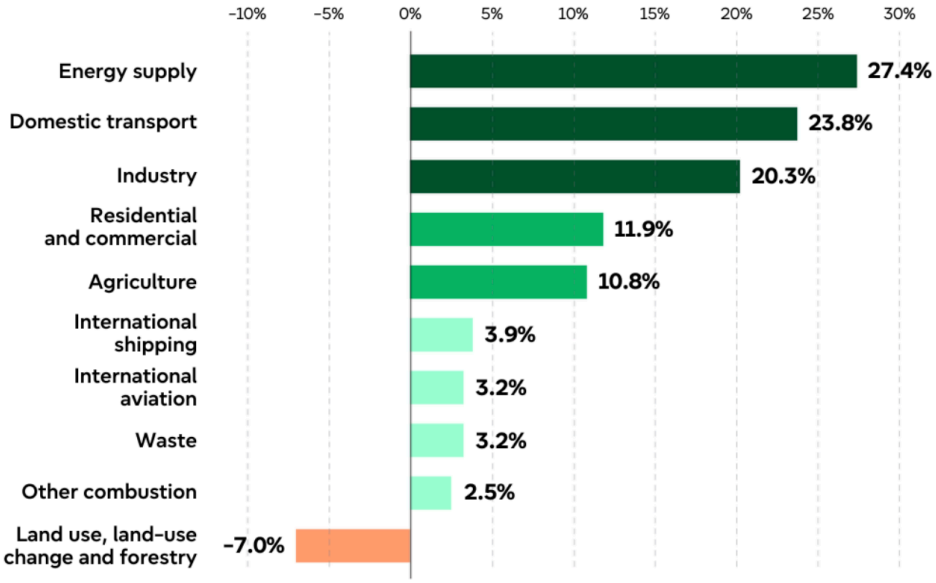


Figure 2.2: EU GHG by Sector [91]

In Figure 2.2, it is clear that the production of energy is the most harmful to the environment. Furthermore, although the aviation sector is not one of the largest contributors to GHG emissions at 3.2%, it is still an industry that requires improvement to prevent the drastic consequences that will follow if left untreated.

2.2. Energy Sector

The energy sector, as clearly shown in Section 2.1, is responsible for the largest portion of GHG emissions, as the most widespread sources of energy are coal, oil, and gas, which are all GHG emitters [103]. The global energy consumption by source is presented in Figure 2.3.

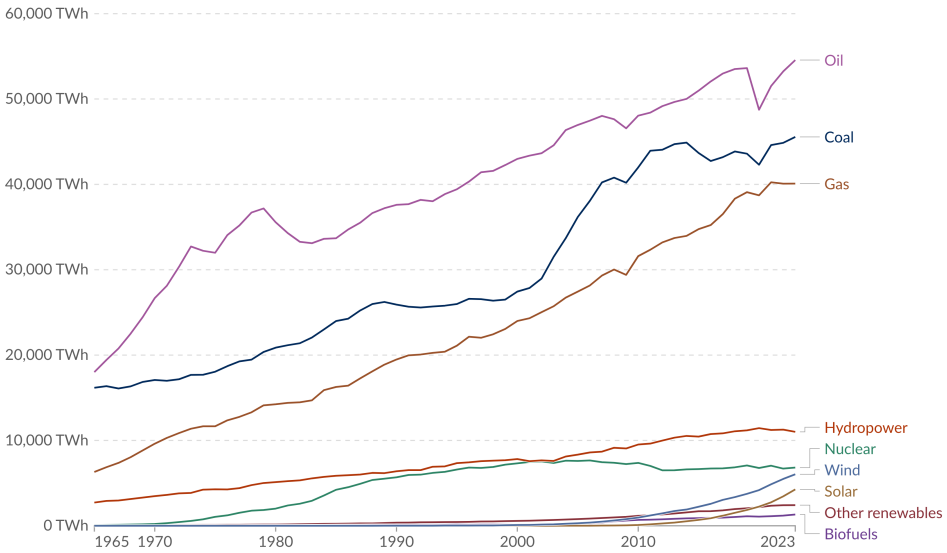


Figure 2.3: Energy Source Usage [103]

Even though attempts are being made to increase the renewable energy share in the energy sector, oil, coal, and gas are much more prominent. The EU has made major strides in diversifying its energy sources; even though developing nations such as China and India have made investments in cleaner sources, they are more reliant on cheaper and easier methods of energy production such as oil, coal, and gas [35].

Expanding further on gas consumption, this can be done through land-based gas turbines. These are gas turbines that are purely created for power generation, that may be used for factories, plants, or to provide electricity for living. Notable heavy-duty gas turbine producers are Siemens, General Electric, Ansaldo, Mitsubishi, etc [19]. However, these gas turbines also contribute to GHG emissions. The European Bureau for Research on Industrial Transformation and Emissions performed a study to obtain the data relating to GHG emissions of combustion plants per country in 2013. This data is given in Figure 2.4.

	Number of plants	MW_{th}	SO_2 (t)	NO_X (t)	Dust (t)
AT	90	19 709	2 718	6 478	456
BE	86	20 367	2 218	9 558	173
BG	24	23 601	116 666	36 846	4 479
CY	16	3 864	10 396	2 908	345
CZ	101	43 639	98 321	65 802	3 389
DE	560	272 785	160 299	226 042	5 385
DK	73	16 481	3 353	8 533	644
EE	20	10 300	25 992	9 430	7 866
EL	50	24 613	52 736	39 633	12 403
ES	144	77 686	92 841	88 091	4 816
FI	159	30 493	19 308	30 757	1 062
FR	235	79 595	79 640	57 680	4 311
HR	13	4 617	6 925	7 470	109
HU	42	17 665	8 627	12 922	424
IE	27	13 973	10 188	9 245	388
IT	342	136 366	29 963	49 695	1 280
LT	21	12 488	2 094	2 250	138
LU	1	730	2	175	0
LV	20	5 711	49	1 054	2
MT	9	1 745	4 880	2 954	228
NL	146	50 529	9 681	21 367	355
PL	96	104 409	324 712	219 905	15 490
PT	26	12 387	5 760	8 435	288
RO	85	36 459	160 211	42 065	10 007
SE	128	28 340	1 794	6 314	449
SI	16	4 653	5 292	8 430	269
SK	63	11 286	40 076	11 978	706
UK	248	152 905	152 644	198 000	6 947
Source: [113, EEA 2013]					

Figure 2.4: GHG Emission and Power Statistics per Country in the EU, 2013 [74]

At 198,000 tonnes of NOx produced in a single year, although gas turbines are regarded as highly efficient, improvements are still needed.

2.3. Aerospace Sector

Although the first airplane was invented in 1903, the aviation industry has expanded dramatically over the past century. The use of combustion for propulsion enabled flights exceeding 10,000 km, but its drawbacks were largely overlooked until the impacts of climate change became evident. Section 2.1 stated that international aviation was responsible for merely 3.2% of global GHG emissions in 2012 in the EU. The drastic measures imposed highlight the EU's determination to address even comparatively minor sources of emissions.

Global CO_2 emission in the aviation industry seems to steadily increase, as a study by Bergero et al. in the website Our World in Data [102] dictates. The rise of CO_2 in aviation is present in Figure 2.5.

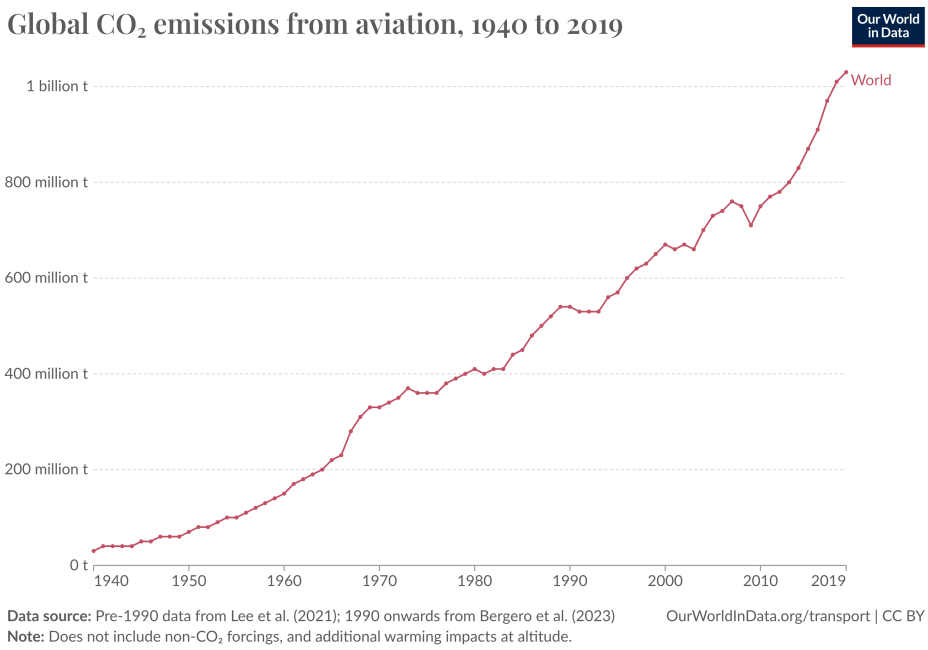


Figure 2.5: Global CO_2 Emissions from Aviation [102]

Aviation emissions have a disproportionately high impact compared to other sources due to the altitude at which they are released. Aircraft operate in the upper troposphere, where GHGs contribute to radiative forcing, amplifying their environmental effects [114]. The Intergovernmental Panel on Climate Change (IPCC) introduced the Radiative Forcing Index (RFI), a multiplier applied to CO_2 emissions to more accurately quantify their true atmospheric impact [49].

On top of this, many countries agreed on a set baseline for the Carbon Offsetting and Reduction Scheme for International Aviation (CORSIA) . Starting in 2024, emissions from international aviation must not exceed 85% of their 2019 levels, as enforced by the International Energy Agency (IEA)[62].

Furthermore, one must consider the efficiency requirements that are set by the EU on the combustion of natural gas. These are enforced on the efficiency level of the best available technique, mentioned as BAT-AEEL (best available technique-an energy efficiency level) . The requirements are given in the Official Journal of the EU [29] and are shown in Figure 2.6.

Type of combustion unit	BAT-AEELs (1) (2)				
	Net electrical efficiency (%)		Net total fuel utilisation (%) (3) (4)	Net mechanical energy efficiency (%) (4) (5)	
	New unit	Existing unit		New unit	Existing unit
Gas engine	39,5–44 (6)	35–44 (6)	56–85 (6)	No BAT-AEEL.	
Gas-fired boiler	39–42,5	38–40	78–95	No BAT-AEEL.	
Open cycle gas turbine, ≥ 50 MWth	36–41,5	33–41,5	No BAT-AEEL	36,5–41	33,5–41
Combined cycle gas turbine (CCGT)					
CCGT, 50–600 MW_{th}	53–58,5	46–54	No BAT-AEEL	No BAT-AEEL	
CCGT, ≥ 600 MW_{th}	57–60,5	50–60	No BAT-AEEL	No BAT-AEEL	
CHP CCGT, 50–600 MW_{th}	53–58,5	46–54	65–95	No BAT-AEEL	
CHP CCGT, ≥ 600 MW_{th}	57–60,5	50–60	65–95	No BAT-AEEL	

Figure 2.6: BAT-AEEL of Natural Gas Combustion [29]

These efficiency levels would also force gas turbine constructors to use smarter techniques to increase efficiency and decrease emissions. Some techniques are proposed in the Journal such as advanced control system, water/steam addition, dry low-NOx burners (DLN), low-NOx burners (LNB), selective catalytic reduction (SCR), etc.

2.4. Gas Turbines

The usage of gas turbines has been prominent in both the energy and the aerospace sectors due to their ease of operation and relatively high efficiency.

2.4.1. Current Constructors

Most notably, Siemens is a very large producer of industrial gas turbine engines. Their SGT series of gas turbines comprises multiple gas turbines with different power outputs depending on demand. To increase efficiency, most of the engines are compatible in combined cycle processes [113]. This involves using the exhaust gas of the gas turbine engine to drive a heat recovery steam generator (HRSG) to use the heat in the exhaust gas to heat steam. This steam is then converted to electricity from the steam turbine [121]. Moreover, the engines are also capable of using certain BATs listed in Section 2.3 such as water/steam addition, SCR, and DLN [113]. For example, their largest turbine (SGT5-9000HL at 593MW) can lower the NOx emission to 2 parts per million per volume, dry (ppmvd) and 10 ppmvd of CO using SCR.

Importantly, the Siemens engines are also equipped to combust hydrogen as well either using DLN or wet low-NOx (WLN) burners. The capabilities of the Siemens gas turbines are given in Figure 2.7.

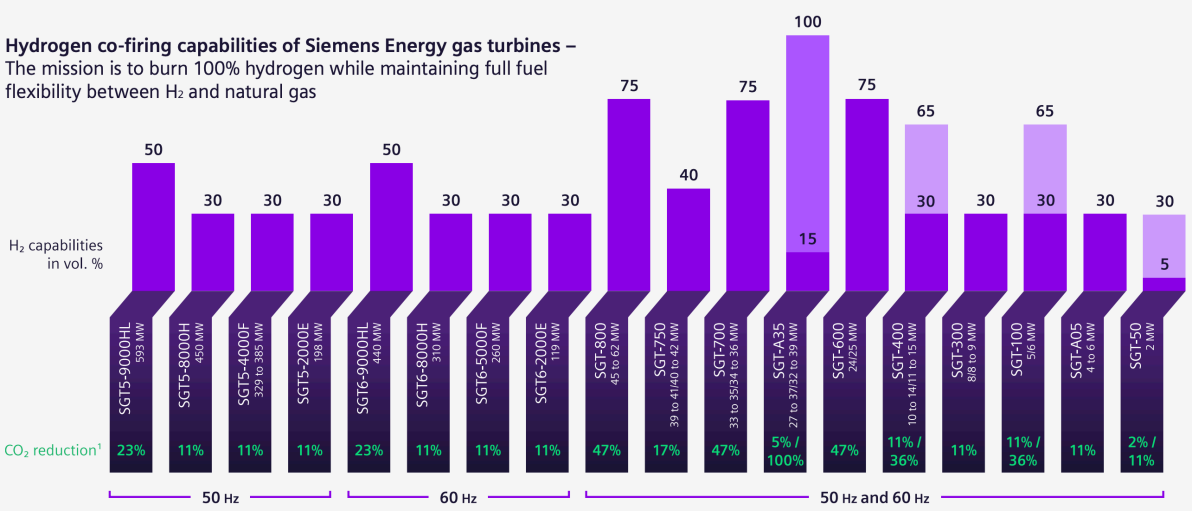


Figure 2.7: Siemens GT H_2 Combustion Capabilities [113]

The reason for this movement towards renewable fuels is due to the attempt of decarbonizing fuels, and using sustainable and renewable methods of combustion.

Other large companies like GE Vernova are also making investments into gas turbines with hydrogen combustion capabilities. Currently, their HA (heavy duty) gas turbines can burn up to 50% hydrogen by volume [122] by also using BAT techniques such as DLN, depicting the movement of fuels towards sustainability. Their plan for HA gas turbines is also visible in Figure 2.8.

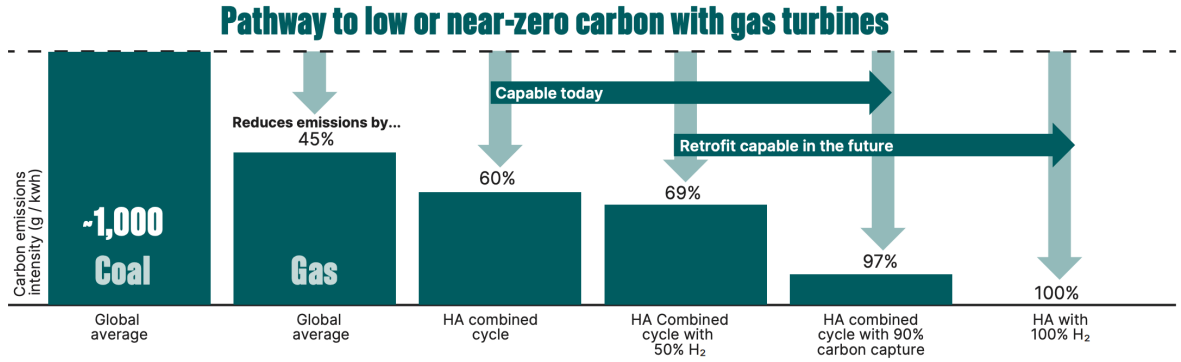


Figure 2.8: GE Vernova's Movement towards H_2 [123]

Alongside GE Vernova, Mitsubishi has successfully operated its M501J gas turbine with a 30% hydrogen blend. The company is now advancing the development of a DLN system for 100% hydrogen firing, with rig tests expected to be completed by March 2025 [88].

2.4.2. Reheat Combustor

A reheat combustor will be used in this work, to perform the necessary studies and assess the research questions. Of interest is Ansaldo Energia, which has been developing single stage and sequential combustors that can combust mixes of hydrogen. These are shown in Figure 2.9.

Technology	Application in Gas Turbine (No hardware modification on gas turbine)	H_2 Capability: any blend between 0 up to max [vol %]
Sequential Combustion	GT36 New and Service	70
Sequential Combustion	GT26 New and Service	45
Single Stage Combustion	AE94.3A New and Service*	40
Single Stage Combustion	AE94.2 New and Service*	40
Single Stage Combustion	AE64.3A New and Service	40

**including V94.3A/V94.2 technology*

Figure 2.9: Ansaldo Energia Engine Types [37]

Of these engines listed, the Ansaldo Energia GT36 depicts the best capability of combusting H2, by using sequential combustion. Not only this, but recent reports from Ansaldo indicate that the engine has been tested with 100% hydrogen, marking a very important milestone [9]. However, the primary challenge of hydrogen combustion stems from its high reactivity, which increases the risk of flashback. Unlike natural gas, hydrogen flames tend to shift upstream, making conventional combustion systems struggle to accommodate hydrogen's characteristics without sacrificing performance. Reducing fuel injection helps mitigate flashback and stabilizes the flame position, but it also lowers exit temperatures, significantly impacting efficiency. Furthemore, the GT36 shows cases of flashback using a trapped vortex combustion system, which makes it an ideal engine to use as a test case.

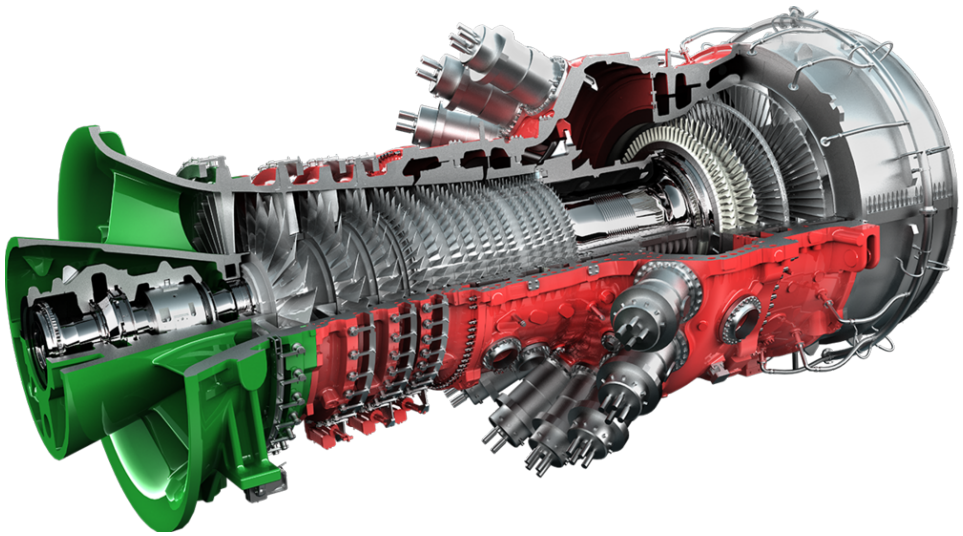


Figure 2.10: Ansaldo Energia GT36 [36]

The GT36, shown in Figure 2.10, employs a sequential combustion system with two lean premixed combustor stages to address this. The first stage stabilizes the flame using flame propagation assisted by a vortex breakdown mechanism, while the second stage (or reheat stage) relies on autoignition for stabilization. This design allows flexible fuel injection across both stages, making it particularly advantageous for high-reactivity fuels like hydrogen.

In the first stage, fuel injection is reduced, which moves the flame position downstream, avoiding burner wall overheating and decreasing residence time. This, combined with a lower equivalence ratio, results

in a lower flame temperature. These factors contribute to a reduction in NOx emissions via the thermal or Zeldovich pathway, where reactions such as

$$\begin{aligned} O_2 + N &\rightleftharpoons NO + O \\ O + N_2 &\rightleftharpoons NO + N \\ N + OH &\rightleftharpoons NO + H \end{aligned} \tag{2.1}$$

are suppressed [120]. Lower residence time and reduced kinetic energy of molecules further decrease NOx formation by limiting high-activation-energy reactions. The unused fuel from the first stage is redirected to the second stage, where autoignition sustains combustion. Here, the flame position is independent of fuel quantity and is determined by the inlet temperature instead. The lower mean exit temperature (MET) of the first stage, combined with air dilution between the stages, decreases the inlet temperature of the second stage, pushing the flame downstream due to an increased ignition delay time. This reduction in residence time further limits NOx formation. Additionally, the presence of water vapor from the first stage further reduces NOx emissions by influencing equilibrium reactions. According to Le Chatelier's principle, added water suppresses reactions such as

$$\begin{aligned} 2\,OH &\rightleftharpoons H_2O + O \\ OH + H_2 &\rightleftharpoons H_2O + H \end{aligned} \tag{2.2}$$

while increasing the formation of H_2O and the reduction of OH radicals, thus limiting the availability of O and OH species necessary for NOx production [20] [42].

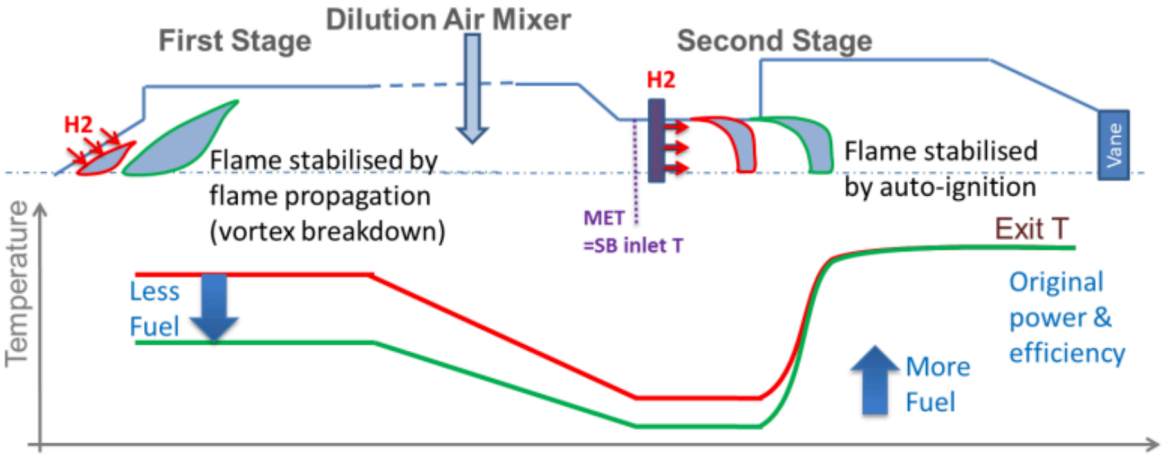


Figure 2.11: Sequential Combustor Working Principle [20]

Figure 2.11 illustrates how these adaptations allow hydrogen integration while maintaining power and efficiency. The red line represents the baseline case, while the green line shows the optimized system with hydrogen. It is worth noting that in the GT26, sequential combustion consists of two combustion stages separated by a high-pressure turbine. The first stage operates at pressures exceeding 30 bar, while the second stage functions at approximately half of the first stage pressure. In contrast, the GT36 does not include a high-pressure turbine but still retains the advantages of the sequential combustion concept. In this design, both combustion stages operate at similar pressures, following the Constant Pressure Sequential Combustion (CPSC) principle [27]. Therefore, the possibility of hydrogen flashback should be investigated, especially considering the higher pressure of the reheat combustion chamber.

3

Hydrogen and Water Applications in Engines

This chapter explores the application of hydrogen and water. It begins by comparing hydrogen's combustion properties with methane and highlights its potential as a low-emission fuel. Key instabilities such as thermoacoustic effects, flashback, and autoignition are examined, with a focus on their impact in reheat combustors like the GT36. Following this, the role of autoignition is analyzed through ignition delay times and explosion limits. The final section discusses water injection as a method to suppress flashback and reduce emissions. Spray characteristics, injection parameters, and timing strategies are reviewed to assess their effectiveness in hydrogen-fueled systems.

3.1. Hydrogen

Using a sequential combustor such as the Ansaldo Energia GT36 mentioned in the previous chapter can create opportunities to research the place of hydrogen as a sustainable fuel option, also due to the ease of production. Hydrogen can be produced using various methods, which give the associated hydrogen product a specific color to describe its nature. These are summarised in Figure 3.1.

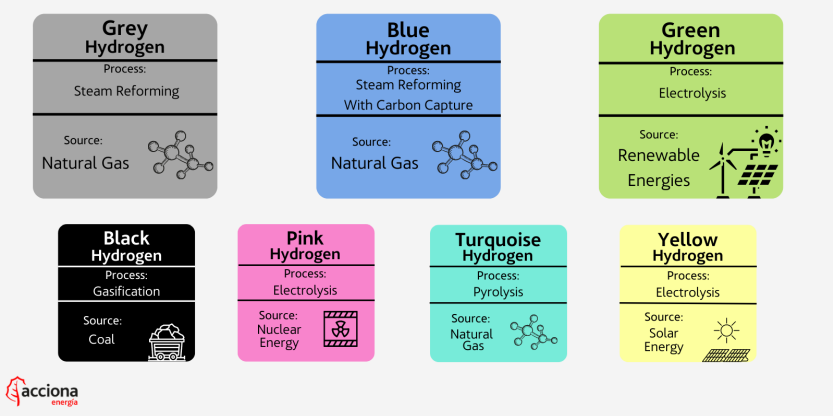


Figure 3.1: Hydrogen Production Methods [3]

Naturally, green hydrogen is preferred due to its clean method of production. However, even if green

hydrogen can be produced, its role in modern combustion must be studied.

3.1.1. Hydrogen Combustion

Hydrogen, as a fuel, has been an intriguing option for decades. Due to its high gravimetric density, renewable nature and potential for cleaner emissions, it is currently regarded as the future of fuels compared to other types, as shown in Figure 3.2.

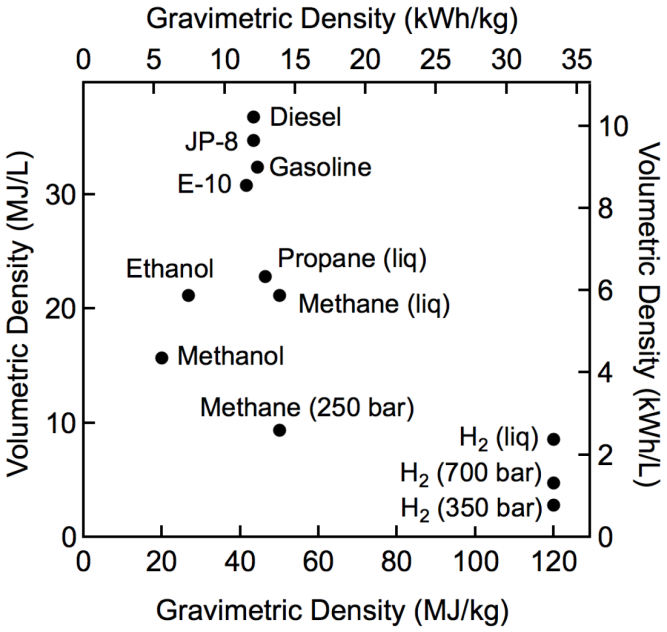


Figure 3.2: Hydrogen Energy [38]

Although the gravimetric density is high, its volumetric density is much lower than other conventional fuels, even when compressed or cooled to a liquid state. On top of this, extra infrastructure is required to ensure that the hydrogen is stored properly in a tank that can sustain its compressed or liquified state, incurring extra weight. However, under the right conditions, the absence of carbon molecules in hydrogen can produce cleaner emissions during combustion.

In Table 3.1, hydrogen's flammability characteristics are compared with methane, a commonly used fuel in industrial gas turbines.

One of the most notable differences between these fuels is their energy content, represented by the LHV. Hydrogen has an LHV of 119.93 MJ/kg, more than twice that of methane at 50.02 MJ/kg [117]. However, hydrogen's low density of 0.090 kg/m^3, compared to methane's 0.716 kg/m^3 [117], presents challenges in fuel injection, storage, and mixing. The stoichiometric air-fuel ratio of hydrogen is also higher at 34.3, nearly double that of methane at 17.23 [57]. This means that hydrogen combustion requires a significantly larger volume of air to achieve complete oxidation.

In terms of flame characteristics, hydrogen exhibits an adiabatic flame temperature of 2376 K, which is higher than methane's 2223 K [13]. In a sequential combustor, this higher flame temperature can lead to increased thermal efficiency but also raises concerns regarding NOx emissions and material limitations. Additionally, hydrogen has an extensive flammability range, spanning from 4% to 75% by volume in air, whereas methane's range is much narrower at 5.3% to 15% [13]. While this broad range offers flexibility in lean-burn operation, it also increases the risk of unintended ignition within the combustor.

Table 3.1: Hydrogen Combustion Properties - *At 20°C and 1 bar

	Hydrogen	**Methane**
Density [kg/m^3]	0.090	0.716
Lower Heating Value (LHV) [MJ/kg]	119.93	50.02
Stoichiometric Air-Fuel Ratio	34.3	17.23
Adiabatic Flame Temperature* [K]	2376	2223
Flammability Limits (vol$\%$ in air)	4-75	5.3-15
Autoignition Temperature [K]	858	813
Minimum Ignition Energy* [mJ]	0.02	0.29
Laminar Flame Speed* [m/s]	3.06	0.376
Diffusion Coefficient in Excess Air [cm$^2/$s]	0.7879	0.2398

Hydrogen's autoignition temperature of 858 K is slightly higher than methane's 813 K [13], making hydrogen more resistant to autoignition in a sequential combustion environment. However, this still requires careful management to avoid premature ignition, especially in the mixing section. Additionally, hydrogen's minimum ignition energy is just 0.02 mJ, significantly lower than methane's 0.29 mJ [13], highlighting its extreme sensitivity to ignition sources. While beneficial for ensuring reliable ignition, this also poses safety and stability concerns.

A large challenge in using hydrogen in a sequential combustor is flashback prevention. Hydrogen's maximum laminar flame speed is 306 cm/s, much higher than methane's 37.6 cm/s [13], meaning that the flame can propagate upstream more easily. Furthermore, hydrogen has a much shorter quenching distance of 0.51 mm [22], compared to 2.5 mm for methane [26]. This allows hydrogen flames to pass through much smaller gaps, significantly increasing the risk of flashback into premixing zones. More importantly, it can also burn closer to the wall, thus enabling boundary layer flashback.

Lastly, hydrogen's mass diffusivity in excess air is 78.79 mm$^2/$s, significantly higher than methane's 23.98 mm$^2/$s [13]. This higher diffusivity enhances fuel-air mixing, which can contribute to more uniform combustion and reduced emissions. However, it also affects flame anchoring and stability, requiring careful control of local equivalence ratios. From these parameters, it can be concluded that while hydrogen combustion is promising, it is also prone to instabilities such as flashback and uncontrolled ignition.

3.1.2. Hydrogen Instabilities

In combustion, hydrogen's instabilities make it difficult for hydrogen to be integrated into today's gas turbines. The main instabilities are summarised below.

Thermoacoustic instabilities are a critical concern in combustion research, characterized by large amplitude oscillations of acoustic modes in a combustor, driven by the interaction between oscillatory flow and unsteady heat release processes. When left unchecked, these instabilities can cause significant issues such as vibrations in components, increased heat transfer rates, flame blow-off, and flashback. Over time, the oscillations can damage the system, limiting the engine's operating envelope or even causing structural failures. The onset of self-sustaining combustion-driven oscillations depends on the phase relationship between heat release fluctuations and pressure oscillations. If they are in phase (within 90 degrees), instability may occur. Moreover, the heat release fluctuations must transfer energy to the unstable acoustic modes faster than energy is dissipated, as formulated by the Rayleigh

criterion. The oscillations grow exponentially at first before reaching a limit cycle, where the amplitude stabilizes due to nonlinear effects [13].

Thermo-diffusive instabilities arise due to differential diffusion of heat and species, particularly in lean hydrogen flames where the Lewis number (Le) of hydrogen is less than 1. This means mass diffusion is faster than thermal diffusion. These instabilities amplify small perturbations in the flame front, leading to the formation of cellular structures and flame fingers. The instability is driven by the low Le of hydrogen, which causes strong differential diffusion effects within the flame front. When thermo-diffusive instabilities are present, the flame exhibits a wide range of unstable wave numbers, resulting in significant wrinkling of the flame front and enhanced flame propagation [18].

Hydrodynamic instabilities (Darrieus-Landau instability) are caused by the density change across the flame front. This instability is present in all premixed flames and tends to form large-scale cusps and fractal-like flame structures. Unlike thermo-diffusive instabilities, hydrodynamic instabilities do not have an intrinsic length scale, leading to large-scale corrugations without the formation of small cellular structures [18].

The thin flame front of hydrogen and its low Lewis number make it challenging to maintain stable combustion. This characteristic often leads to a common issue in hydrogen combustion known as flashback.

3.1.3. Flashback

Flashback remains a critical challenge in gas turbine design, often necessitating significant modifications to ensure safe and efficient operation. Its occurrence can compromise both system reliability and performance, leading to severe safety hazards and operational inefficiencies. The primary mechanisms responsible for flashback are outlined below [15].

Combustion instabilities, as discussed in subsection 3.1.2, may arise if the combustion process induces large pressure fluctuations within the combustor. Sustained operation with high pressure fluctuations must be avoided, as the combustor system may suffer damage due to fatigue. However, transient high-pressure fluctuations are common. These transient events can lead to extremely low flow velocities, allowing the flame to propagate deep into the burner due to core flow and/or wall boundary layer flashback. Although propagation is the mechanism leading to flashback in this case, combustor damage occurs over a finite time, typically much longer than the duration of such transient events. Thus, damage can be prevented if, after such an event, the flame moves back into the combustor and does not remain in the burner. The avoidance of combustor damage relies on flame quenching and extinction.

Flame propagation in the core flow may occur if there is an increase in the turbulent flame velocity or a decrease in the flow velocity. The former can result from an increase in flame temperature or fuel reactivity due to changes in fuel composition. Using hydrogen, which is known to have a higher flame velocity, is even more risky for flashback. Experimental studies provide insights into flame propagation under different conditions, but variations in pressure, temperature, and fuel composition can lead to discrepancies between predictions and observed behavior.

Flame propagation within boundary layers may occur for similar reasons as core flow propagation. Boundary layer flashback occurs when a flame propagates upstream into the boundary layer of a gas turbine burner. This happens when the local flame speed exceeds the flow velocity within the boundary layer, particularly near the wall where the velocity decreases due to the no-slip condition. However, the likelihood of flashback is influenced by factors such as flame quenching due to heat loss to the wall and flame stretch effects, which can counteract its occurrence. In the case of laminar boundary layer flashback, the critical velocity gradient concept is used to describe the flame's stabilization at a penetration depth where the local velocity matches the flame speed. The critical velocity gradient

depends on various factors, including fuel composition, temperature, pressure, and wall conditions. Fuels like hydrogen significantly increase the risk of flashback due to their high flame speeds, making them more challenging to control in gas turbine applications. Experimental studies, such as those conducted by Dam et al. [32], have shown that blends with higher hydrogen fractions exhibit stronger flashback tendencies.

Combustion-Induced Vortex Breakdown (CIVB) While the previously mentioned mechanisms apply to any premixed burner, CIVB is specific to swirl-stabilized burners (not the case of the GT36). In this scenario, the combustion process alters the burner fluid dynamics, causing the vortex breakdown bubble to shift from the burner exit region to deep within the burner. One of the key mechanisms responsible for this phenomenon is the misalignment between surfaces of constant pressure, generated by the swirling flow, and surfaces of constant density, generated by combustion. This misalignment induces baroclinic torque, which promotes negative velocity along the burner axis, further driving the vortex breakdown process.

Autoignition flashback

In practical systems, this phenomenon can arise due to a reduction in flow velocity or an increase in the temperature of the fuel/air mixture. The temperature rise of the premixed fuel may be caused by convective heating from the burner surfaces, which, in turn, can be heated by radiative heat feedback from the combustor. Recent studies (Fritz et al. [44]) have shown that autoignition may not be considered a form of flashback due to the mechanism that causes it not to be related to flame propagation. However, autoignition is accepted as flashback, as its consequences are similar to the flame propagation of flashback mechanisms. As this is the prevailing flashback mechanism in the Ansaldo Energia GT36 [104], this mechanism will be investigated further.

3.1.4. Autoignition

Autoignition refers to the spontaneous ignition of a combustible mixture due to the thermodynamic conditions of the system. This process occurs when reactions are initiated and generate enough heat to sustain combustion without an external ignition source. The onset of autoignition also requires a certain amount of time, known as the autoignition delay time τ_{ig}. This delay time can vary significantly, from hundreds of microseconds to several seconds, depending on the initial thermodynamic state of the mixture.

In a sequential combustor system like the GT36, the combustor is designed for autoignition. The geometry of the reheat combustor must be built in a way that allows the time for mixing of the premixed fuel to be completely homogeneous; however, the residence time in the mixing duct has to be shorter than the τ_{ig} to avoid premature autoignition. Furthermore, the conditions inside the combustor must be ideal for the hydrogen to autoignite as safely and efficiently as possible, due to autoignition delay time being influenced by temperature, pressure, and equivalence ratio. This is visible in Figure 3.3.

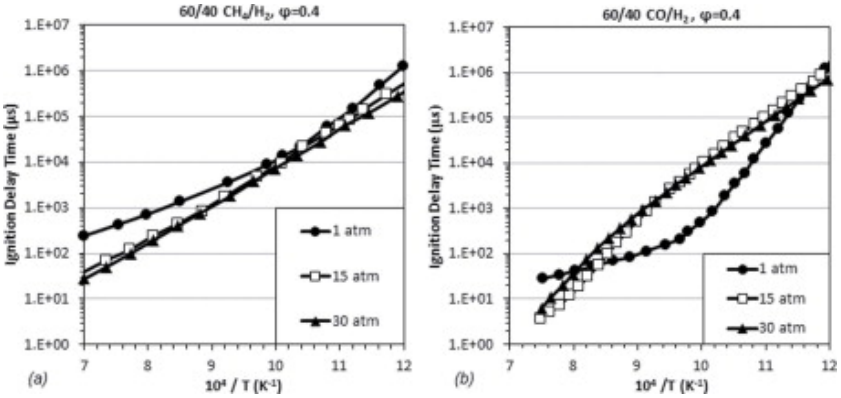


Figure 3.3: Pressure and Temperature Influence on τ_{ig} of Different Fuel Compositions [17]

The findings indicate that contrary to what would be predicted, ignition delay time does not always decrease as pressure rises. For example, at 1000 K, the ignition delay time for the CH4/H2 blend Figure 3.3a is nearly constant with pressure. In the CO/H2 mixture, the impact of pressure is considerably more noticeable (Figure 3.3b). The ignition delay durations at 1 atm are expected to be shorter than those at 15 and 30 atm for temperatures ranging from around 1200 to 900 K. The anticipated ignition delay times at 15 atm are, however, less than those at 30 atm at higher temperatures. Regarding equivalence ratio, Benim and Syed [15] state that ignition delay time shows a mild increasing behavior with an increasing equivalence ratio. It should be noted that in both subfigures, it is clear that the trends are not linear, and that a slope is present, implying that there is a heavier dependency on temperature in some regions. As the temperature is further increased past the bounds of Figure 3.3, local minima of the τ_{ig} are present, which correlate to the cross-over temperature, which will be explained further. These minima also depict a behavior known as explosion limits.

The autoignition behavior of hydrogen is governed by three accepted explosion limits as discussed by Sánchez, Fernández-Tarrazo, and Williams [109], which delineate the boundary between explosive and non-explosive regimes. These limits arise from the interplay between radical chain branching reactions, diffusive transport of reactive species, and radical termination mechanisms. The pressure-temperature graphs exhibit an S-shaped explosion boundary, whose structure is determined by these fundamental processes. An example of this is shown in Figure 3.4.

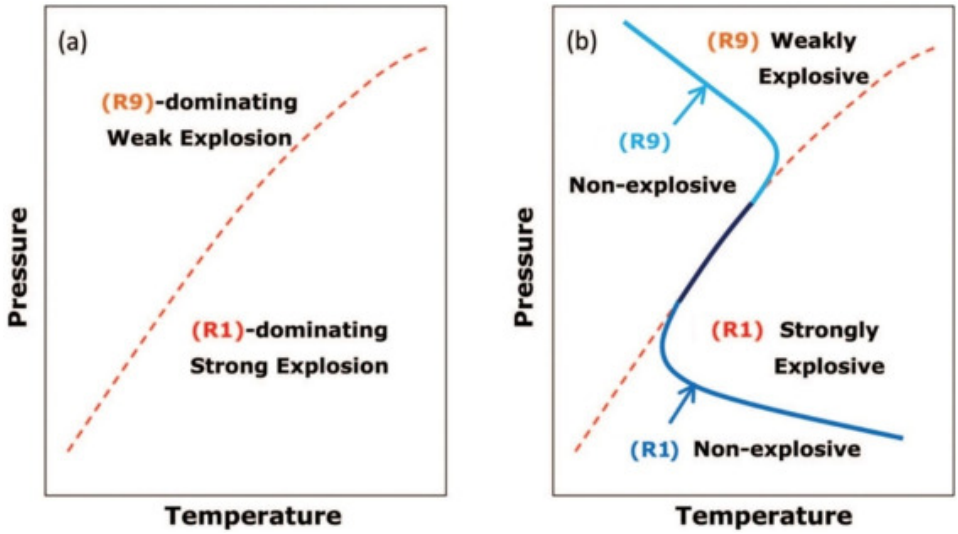


Figure 3.4: a) Explosion Limit of Hydrogen without Wall Deactivation b) with Wall Deactivation [80]

The first explosion is controlled by diffusion. At low pressures, the radical chain branching mechanism, primarily governed by the reaction

$$\mathrm{H} + \mathrm{O_2} \longrightarrow \mathrm{OH} + \mathrm{O} \tag{3.1}$$

initiates combustion. The newly formed radicals participate in additional reactions such as:

$$\mathrm{H_2} + \mathrm{O} \longrightarrow \mathrm{OH} + \mathrm{H} \tag{3.2}$$

$$\mathrm{H_2} + \mathrm{OH} \longrightarrow \mathrm{H_2O} + \mathrm{H} \tag{3.3}$$

The net effect is the overall chain-branching reaction:

$$3\,\mathrm{H_2} + \mathrm{O_2} \longrightarrow 2\,\mathrm{H_2O} + 2\,\mathrm{H} \tag{3.4}$$

Specifically Figure 3.4a shows that in the absence of wall deactivation of the radicals, the system is explosive in the lower pressure regime, being controlled by the two-body branching reaction Equation 3.1. Figure 3.4b shows that in the presence of a wall, the diffusion of radicals (H, O, OH) to the chamber walls outpaces their generation in the gas phase. Since radical recombination on the chamber walls effectively removes active species, chain branching is suppressed, preventing an explosion. Given that molecular diffusivity is inversely proportional to pressure, as the pressure increases, diffusive losses become less significant, eventually allowing the reaction to self-sustain.

For the second explosion limit, as the pressure increases further, the radical destruction mechanism becomes dominated by three-body termination reactions, particularly:

$$\mathrm{H} + \mathrm{O_2} + \mathrm{M} \longrightarrow \mathrm{HO_2} + \mathrm{M} \tag{3.5}$$

where M represents a third-body collision partner. Equation 3.5, also known as R9, produces the weakly-reactive HO2 radical, which becomes progressively more important and leads to the transition to a weakly explosive regime. The formation of hydroperoxyl radicals (HO_2) results in radical recombination, thereby reducing the overall concentration of chain carriers. The effectiveness of this termination pathway increases with pressure due to the higher frequency of three-body collisions.

The second explosion limit is thus characterized by the competition between chain-branching and chain-terminating reactions. When termination dominates, radical production is insufficient to sustain combustion, leading to a non-explosive regime. The transition temperature at which the radical branching and termination rates are exactly balanced is known as the *crossover temperature*. Below this temperature, termination is favored, preventing explosion, whereas, above this temperature, branching dominates, leading to ignition. As an example, The change from the first to the second explosion limit is seen by the shape of the 1 atm curve of Figure 3.3.

Finally, at the third explosion limit (very high pressures), combustion transitions from a chemically controlled to a thermally controlled explosion regime. In this regime, the dominant radical chain carriers shift from H, O, and OH to HO_2 and H_2O_2, with key reactions including:

$$\mathrm{H} + \mathrm{O_2} + \mathrm{M} \longrightarrow \mathrm{HO_2} + \mathrm{M} \tag{3.6}$$

$$2\,\mathrm{HO_2} \longrightarrow \mathrm{H_2O_2} + \mathrm{O_2} \tag{3.7}$$

$$\mathrm{H_2O_2} + \mathrm{M} \longrightarrow 2\,\mathrm{OH} + \mathrm{M} \tag{3.8}$$

Unlike in the first and second explosion limits, where radical transport and chemical kinetics determine ignition, in the third limit, the explosion is driven by a thermal runaway process. The decomposition of hydrogen peroxide releases heat, which in turn accelerates reaction rates, ultimately leading to a self-

sustaining thermal explosion. At sufficiently high pressures, the heat release rate exceeds conductive heat losses, resulting in an exothermic runaway. The third explosion limit therefore represents a transition from kinetic control to thermal ignition, where heat accumulation dictates the onset of explosion. It is good to consider that Rouco Pousada et al. [104] determined that at T = 1180 K, the chemical kinetics of the GT36 fall within the second ignition limit. This analysis offers crucial insights into the species and reactions to focus on when investigating flashback precursors.

Naturally, autoignition is also affected by flow properties such as turbulence. Gruber et al. [51] performed a DNS revealing that turbulence intensity has a direct and pronounced effect on the flame front velocity, visible in Figure 3.5a. As the turbulence intensity increases, the flame front velocity also increases. This acceleration is primarily attributed to the enhanced mixing of the reactant gases and the associated increase in the rate of chemical reactions. The increased turbulence promotes more efficient transport and distribution of reactants, leading to a faster and more uniform combustion process. The relationship between turbulence and flame front velocity in hydrogen combustion is particularly significant due to the high diffusivity and reactivity of hydrogen, which can amplify or suppress turbulent flame propagation depending on the combustion regime.

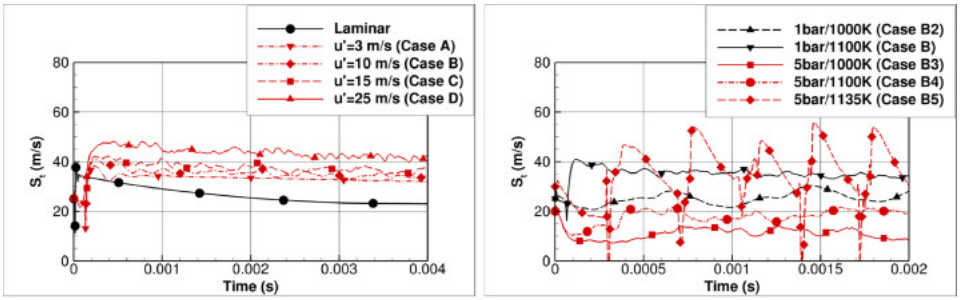


Figure 3.5: Turbulent Flame Speed of Hydrogen affected by a) Flow Turbulence b) Temperature/Pressure

Additionally, the study explores the influence of compressibility effects on flame stability under turbulent combustion conditions. It was found that compressibility significantly impacts the behavior of the flame front, particularly at high turbulence levels. The combustion system exhibited unstable ignition and flame behavior, especially when the reactant temperature approached the crossover temperature of the mixture.

Finally, understanding autoignition requires comprehension of the mechanisms. Knowing explosion limits and chemical kinetics allows for a more refined definition of autoignition; when a fuel-oxidizer mixture reaches a critical state where radical production exceeds radical loss, leading to a self-sustaining combustion reaction. In turbulent environments, autoignition does not occur uniformly but rather at discrete locations, forming ignition kernels. These kernels develop in regions where the local temperature and mixture composition favor radical accumulation.

In a study performed by Echekki and Chen [34], the ignition process is characterized by two main stages:

1. **Induction Stage:** This phase is characterized by radical chain-branching reactions occurring in a thermally frozen flow, meaning that heat release is minimal. The radical pool builds up due to nonlinear chemical kinetics, without significant thermal feedback.
2. **Thermal Runaway and Flame Formation:** Once a critical radical concentration is reached, exothermic reactions accelerate, leading to a rapid temperature rise and the formation of a propagating premixed flame.

Ignition kernels form in regions with low heat and radical dissipation. This condition ensures that radical production outpaces radical loss, allowing the kernel to transition to a fully developed flame. Turbulence also adds complexity to the ignition process by altering the transport of heat and reactive species. Two key effects of turbulence on ignition kernels include:

- **Scalar Dissipation Rate:** High scalar dissipation rates, which correspond to strong mixing, can suppress ignition by enhancing radical and heat losses from the ignition site. Conversely, lower dissipation rates favor ignition by reducing these losses.
- **Strain Rate and Mixing:** Turbulent mixing can modulate the local concentration and temperature fields, influencing where ignition occurs. In some cases, turbulence enhances ignition by increasing fuel-oxidizer contact, while in others, it delays ignition by increasing radical losses.

The influence of turbulence is non-monotonic; moderate turbulence can enhance ignition by promoting mixture homogeneity, whereas excessive turbulence can suppress it by increasing radical loss rates. The complexity of hydrogen combustion is further compounded by various control strategies aimed at modulating flame behavior and emissions. One such approach is water injection, which introduces additional variables affecting flame dynamics, temperature profiles, and chemical kinetics. The use of water in engines spans a range of implementations, from relatively simple techniques to more advanced systems, as discussed in the following section.

3.2. Water Injection

In the 20th century, adding water to combustion chambers gained popularity to temporarily boost power production since it increased the mass flow inside the chamber. However, other side effects that were not previously taken into account, such as the reduction in the maximum combustion temperature and the corresponding decrease in NOx production and emission, have made the approach popular again in recent years [30].

Today, many studies have been performed to understand the effect of water injection. Rrustemi et al. [106] investigated how adding water affects the laminar flame speed of lean hydrogen-air mixtures at elevated pressures. The findings indicate that water addition decreases the laminar flame speed in these mixtures. This reduction is attributed to water's thermal and chemical effects, which include absorbing heat and diluting reactive species, leading to slower combustion reactions. Furthermore, Concetti et al. [30] found that water injection reduces the total burning rate, flame area, and burning rate per unit area due to cooling effects, with a significant impact only when evaporation is intense. Hasslberger et al. [56] also performed a similar study using a DNS to find that the presence of water droplets leads to significant reductions in flame temperature and burning velocity, primarily due to the cooling effect from water evaporation. Notably, the influence of droplet size on the overall burning rate is strongly non-linear, whereas the effect of water loading exhibits a fairly linear relationship. Water injection can have multiple effects on a combustion reaction,

3.2.1. Spray Characteristics

To administer the water into the reheat combustor, sprays will be used. However, many parameters can be adjusted to ensure optimal flashback suppression. These are mentioned below.

1. **Spray Type:** The type of spray, or atomizer type, can have a large influence on the path that the water injection takes. More importantly, an atomizer is also responsible for breaking up the water into droplets, such that the water can spread easier. Many types of atomizer types can be used, although more common ones include the plain orifice/full cone spray and the simplex/hollow cone spray[75]. A plain orifice injects a round liquid jet into the surrounding air, with finer atomization achieved using smaller orifices. However, practical limitations due to clogging restrict the mini-

mum orifice size to approximately 0.3 mm. In a simplex, liquid enters a swirl chamber through tangential holes, creating a core of air or gas that extends from the discharge orifice to the rear of the chamber. The liquid exits as an annular sheet, spreading radially to form a hollow conical spray with angles ranging from 30° to 180°, depending on the application. Finer atomization occurs at higher delivery pressures and wider spray angles. An example is shown in Figure 3.6.

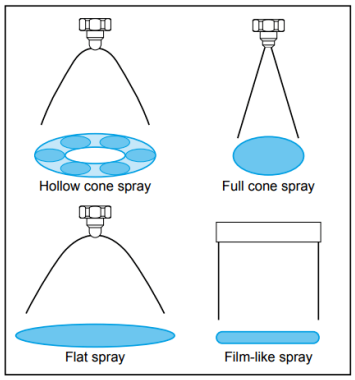


Figure 3.6: Atomizer Types [39]

2. **Residence Time:** Residence time is the time it takes for the water to stay in a certain domain, and is very important for flashback control, efficiency, and emissions. A low residence time leads to low NOx emissions [51], but could also mean lower evaporation efficiency.
3. **Mass Flow and Velocity:** The velocity of the water and its associated mass flow are extremely important, as evidenced by Amani, Akbari, and Shahpouri [7], who found that the maximum temperature in combustion was most sensitive to mass flow rate. Mass flow rate can also influence the trajectory, residence time, evaporation ratio, etc. To estimate the injection velocity for a given mass flow, the following equation can be used [42]:

$$v_{\text{inj}} = \frac{4\dot{m}_L}{\pi d_0^2 \rho_L} \tag{3.9}$$

   where m_L is the mass flow of the liquid, d_0 is the injection diameter and the ρ_L is the density of the liquid.
4. **Sauter Mean Diameter (SMD):** The SMD is the diameter of the sphere that has the same volume to surface ratio as the particles of interest in the spray. It is indicative of the sizes of the particles which affects evaporation, collisions, and residence time. A very small particle may have high evaporation efficiency due to its smaller volume [105], while a larger droplet may be beneficial for dispersion due to it being less affected by the core flow [42]. The SMD can be quantified using [104]:

$$r_{\text{SMD}} = \frac{\sum_{i=1}^{N_{\text{tot}}} N_i r_i^3}{\sum_{i=1}^{N_{\text{tot}}} N_i r_i^2} \tag{3.10}$$

5. **Water Temperature:** Since lower flame temperatures can reduce NOx emissions, excessively high water droplet temperatures may be ineffective in achieving this reduction. Additionally, if the water is too hot in the mixing duct, it may fail to absorb sufficient heat from the hydrogen, increasing the risk of autoignition flashback. Conversely, if the water temperature is too low, it can negatively impact thermal efficiency and shift the flame front closer to the mixing duct.
6. **Swirl Number:** This number can be used to quantify the swirling of the water when using a swirling spray. Swirling is generally advantageous for mixing and ensuring droplet breakup, which

means that the temperature of the core flow will reduce, leading to lower emissions and the chance of flashback. A high swirl is also correlated with higher combustion efficiency [7]. However, using a spray with too high of a swirl number can lead to a flashback of the CIVB kind, shifting the vortex breakdown bubble into the mixing duct [16].

7. **Geometry and Placement:** The geometry of an atomizer is defined by several parameters depending on the configuration. In a solid cone atomizer, key factors include the orifice length, diameter, and cone angle, while a simplex injector also considers the thickness angle, which determines the ratio between the air core flow and the water flow. These parameters influence the Sauter Mean Diameter (SMD) and significantly affect droplet dispersion within the combustion chamber, particularly through the external angle and the thickness of the resulting annular sheet. Additionally, the injection location plays a crucial role in suppressing flashback. While premixing ducts are the most common injection sites, Farokhipour, Hamidpour, and Amani [40] found the best injection location to be at the end of the primary zone within the post-flame region. However, given that autoignition flashback propagates at nearly the speed of sound, positioning the injection point closer to the autoignition zone could reduce response time and enhance temperature control.

Even when all the spray characteristics are set, the timing of the sprays is still something of importance. This process of autoignition flashback begins with the precursor kernels, leading to autoignition in a very small timeframe, in the order of milliseconds. Therefore, the water will spray during this timeframe, but how the tapering will be performed is still yet to be explained. Studies such as Floris [42], Wang et al. [126], and Rouco Pousada et al. [104] have attempted to use water spraying to avoid hydrogen autoignition, but little attention has been paid to the timing of the sprays. There have been studies [5, 125, 69] relating to internal combustion engines where the authors have discussed the timing of the stages within the cylinder to reduce knocking, however, the dynamics within a gas turbine behave much differently.

For instance, Floris [42] demonstrated that water spraying effectively suppressed flashback; however, due to its oscillatory nature, the flashback reappeared shortly after. This behavior was particularly influenced by the rapid water injection rate, which contributed to compression effects on the hydrogen. These compression effects may have played a role in enabling the flashback to return. Instead, implementing a tapering water spray, where the mass flow gradually decreases over time while maintaining efficient evaporation and combustion, could offer a more effective solution.

4

Machine Learning Applications

This chapter introduces the data-driven tools used in the study. Key properties of chaotic time series are reviewed, followed by the preprocessing steps (scaling, optional transforms) adopted for learning. Dimensionality-reduction methods are presented with emphasis on autoencoders, together with regularization and training choices. Clustering and regime labelling strategies are then described, including metrics for evaluation (reconstruction error, lead time, and error rates) that will be used consistently in subsequent chapters.

4.1. Chaotic Time Series

A chaotic dynamic time series is a sequence of data points generated by a deterministic system that exhibits chaotic behavior, meaning the system is highly sensitive to initial conditions, leading to complex and unpredictable patterns over time. Despite this randomness, the system's evolution is governed by underlying rules [71].

- **Deterministic Nature:** The system follows well-defined rules or equations without any random input.
- **Sensitivity to Initial Conditions:** Small differences in initial conditions can result in vastly different outcomes, commonly known as the ”butterfly effect.”
- **Nonlinearity:** The governing equations are nonlinear, meaning the output is not directly proportional to the input.
- **Aperiodicity:** The system does not settle into a repeating cycle and continues to evolve unpredictably.

Many examples of chaotic time series are present in society. Weather systems are chaotic, due to atmospheric dynamics being highly sensitive to initial conditions, making long-term forecasting challenging. Population dynamics such as logistic maps display chaotic behavior under specific conditions, leading to unpredictable population fluctuations. More examples like traffic modeling, stock market information, and solar system dynamics [71] can all fit under the umbrella of a chaotic system [98].

4.1.1. Analysis Methods

Due to the complex nature of chaotic time series, many different types of methods are used to analyze its behavior.

The Lyapunov exponent may be used to understand whether a time series exhibits chaotic behavior. Lyapunov exponents quantify the average exponential rates at which nearby trajectories in a dynamical system diverge or converge. If the largest Lyapunov exponent is positive, it signals that even the smallest differences in initial conditions will amplify exponentially over time; a signature of chaos. The inverse of this exponent, known as the Lyapunov time, provides a characteristic timescale over which predictions of the system's state remain reliable before uncertainties overwhelm the forecast. This means that for chaotic time series, while short-term forecasting might be feasible, long-term predictions are inherently limited by the rapid growth of errors. Foundational studies such as those by Wolf et al. [129] have laid the groundwork for understanding and applying these concepts, and further elaborations in nonlinear time series analysis by Kantz and Schreiber [70] continue to inform current methodologies in quantifying and managing predictability in complex systems.

Moreover, one of the major ways of analyzing a time series is through phase space reconstruction. Phase space representation is a technique that offers a geometric perspective on system behavior by mapping all possible states and their trajectories. This approach is valuable for understanding complex, chaotic, or nonlinear systems where traditional time-domain analysis may be insufficient. In dynamical systems, the phase space is a multidimensional space where each dimension corresponds to one of the system's variables. A point in this space represents a specific state of the system, and its trajectory over time illustrates the system's evolution. This visualization aids in identifying patterns, stability, and the nature of attractors within the system, which are sets of states toward which a system tends to evolve. An advantage is time independence; phase space representations factor out the importance of the time variable, and simplify the analysis of dynamical systems. Furthermore, they allow for the identification of attractors, providing insights into the long-term behavior of the system. Finally, reconstructed phase spaces can help distinguish between deterministic chaos and random noise, improving the reliability of the analysis [23].

Another popular method is recurrence analysis. Recurrence analysis offers a powerful framework for investigating the temporal dynamics of chaotic systems by examining when a system's state recurs over time. This method utilizes recurrence plots, which are two-dimensional visual representations that mark the instances at which the state of a system returns to a previous neighborhood in phase space. By quantifying these recurrences through metrics such as the recurrence rate, determinism, and laminarity, collectively known as Recurrence Quantification Analysis (RQA) , researchers can detect subtle changes in the system's behavior and identify transitions between different dynamical regimes. Such analyses have proven especially valuable in distinguishing deterministic chaos from stochastic noise and in capturing the evolution of complex systems across diverse fields, from physiological signal analysis to financial time series forecasting [85].

Finally, machine learning (ML) approaches have emerged as a powerful tool for analyzing and forecasting chaotic time series, offering a data-driven alternative to traditional analytical techniques. These methods, ranging from recurrent neural networks (RNN) and long short-term memory networks (LSTM) to reservoir computing, are capable of capturing the complex nonlinear relationships inherent in chaotic systems without the need for explicit model formulations. For instance, reservoir computing has been shown to effectively predict the short-term evolution of chaotic dynamics even in the presence of noise, thereby enhancing the ability to model systems where deterministic chaos limits long-term forecast horizons [65]. Similarly, adaptive neural network architectures learn to identify underlying structures and patterns directly from data, improving short-term prediction accuracy despite the rapid divergence of trajectories dictated by the system's positive Lyapunov exponents. Although these machine learning techniques extend the practical forecasting window, the intrinsic sensitivity to initial conditions, as quantified by the Lyapunov time, continues to impose fundamental limits on long-term predictions [81]. ML has advanced significantly in recent years, to the point where it is now commonly used alongside many

established analytical methods. The effectiveness of ML, however, varies depending on the specific approach employed, which will be explored in more detail.

4.2. Machine Learning

In the context of hydrogen combustion in sequential combustors, one of the most critical challenges is the prediction and early detection of autoignition flashback events. As discussed in subsection 3.1.4, these events can occur suddenly, initiated by localized ignition kernels that form in regions where radical buildup, turbulence, and heat release converge. Given the short timescales involved—on the order of milliseconds—there is a need for methods that can identify early precursor states in the combustion chamber before the onset of flashback. These precursors are not easily detectable with traditional thresholding or analytical models, due to the complex and chaotic behavior of the flow.

To address this, machine learning (ML) offers a promising data-driven framework capable of handling nonlinear, high-dimensional time series data. ML is a subset of artificial intelligence (AI) that enables computers and machines to mimic human learning, allowing them to perform tasks independently and enhance their accuracy and performance over time through experience and data exposure [59].

Generally, machine learning can be split into three branches: supervised learning, unsupervised learning, and reinforcement learning. Reinforcement learning has seen some exploration in combustion control and the prediction of extreme events in chaotic time series, but its use remains relatively limited compared to more established methods like supervised or unsupervised learning. Therefore, these two branches will be investigated in this work.

To implement a flashback detection pipeline, two ML architectures are required. The first algorithm will extract a latent reduced representation of the combustion chamber time series—capturing the dominant temporal patterns and state changes. The second algorithm will then operate on this reduced representation to identify precursor states through clustering or classification, depending on data availability.

The studies reviewed in the remainder of this chapter are selected with the aim of identifying methods that could be applied or adapted to this problem. Particular attention is given to works that attempt precursor detection in chaotic or combustion-related systems, allowing parallels to be drawn between their approaches and the objectives of this study.

4.2.1. Supervised Learning

In supervised learning, a labeled training dataset is used to comprehend the connections between input and output data. Training datasets with input data and the associated labels are manually created by data scientists. In real-world use applications, supervised learning teaches the model to apply the appropriate outputs to fresh incoming data. Large datasets are processed by the model's algorithm during training in order to investigate possible correlations between inputs and outcomes. To determine whether the model was successfully trained, its performance is then assessed using test data. The process of testing a model using a different subset of the dataset is called cross-validation.

When training neural networks and other supervised learning models, the most popular optimisation or learning algorithms are those in the gradient descent family, which includes stochastic gradient descent (SGD). The loss function, an expression that quantifies the difference between the model's predicted and actual values, is used by the model's optimisation method to evaluate accuracy. The main indicator of model performance is the gradient, or slope, of the loss function. To minimise its value, the optimisation algorithm moves down the gradient. In order to optimise the model, the optimisation algorithm modifies the model's parameters throughout training [14].

Several studies have employed supervised learning methods to accurately predict combustion conditions. For instance, Abdurakipov et al. [1], Han et al. [54], and Hanuschkin et al. [55] investigated the use of logistic regression for classifying combustion regimes, while Bai et al. [12] and Abdurakipov et al. [1] demonstrated the effectiveness of k-nearest neighbor algorithms in predicting combustion performance. Support vector machines have also been widely explored for their robust classification capabilities, as evidenced by the works of Bai et al. [12], Abdurakipov et al. [1], Han et al. [53], and Han et al. [54]. In particular, Han et al. [54] demonstrated how combustion regimes can be classified from pressure signal time series, a relevant insight for this work given the potential to extract similar features in an unsupervised context. Moreover, Han et al. [54] extended the application of supervised methods by employing Gaussian processes, and González-Espinosa et al. [47] applied linear discriminant analysis to further understand combustion behavior. In addition, artificial neural networks, particularly those based on multilayer perceptron architectures (MLP), have been extensively used across multiple studies (Bai et al. [12]; Abdurakipov et al. [1]; González-Espinosa et al. [47]; Han et al. [53]; Han et al. [54]; Hanuschkin et al. [55]; Yang et al. [133]) to harness complex feature sets extracted from flame images. For example, Bai et al. [12] showed that MLP-based architectures are capable of capturing dynamic flame transitions, suggesting their structure could be repurposed for precursor detection using latent features. Therefore, many studies have attempted to use supervised learning techniques in combustion applications, and even precursor detection.

In recent years, additional studies have further advanced the application of machine learning techniques to forecast and manage chaotic dynamics. For instance, Pathak et al. [92] demonstrated that reservoir computing could predict high-dimensional, spatiotemporally chaotic systems without an explicit model, therefore, underscoring its potential in forecasting turbulent flows. Similarly, Champion, Brunton, and Kutz [24] employed data-driven approaches to uncover the underlying coordinates and governing equations of complex systems, depicting the benefits of ML methods for real-time control and prediction. Complementing these efforts, Xiao, Li, and Zhang [130] proposed an LSTM-based framework specifically designed for the early detection of thermoacoustic instabilities, further illustrating the trend toward utilizing deep learning architectures for online monitoring in combustion processes, especially applicable in this work.

Furthermore, more studies have employed advanced machine learning techniques to analyze pressure time series and predict instabilities in thermoacoustic systems. For instance, Bury et al. [21] analyzed pressure time series of thermoacoustic oscillations using a convolutional neural network LSTM (CNN-LSTM) architecture inspired by critical slowing down theory, demonstrating that such a network can effectively detect the onset of bifurcations—even though it sometimes struggled to correctly classify the bifurcation type. This highlights the potential of deep learning models to extract subtle temporal patterns from pressure time series—an ability that is important to this work's aim of identifying flashback precursors in high-dimensional combustion signals. In another study, Asch et al. [10] compared different neural network architectures, including feedforward, LSTM, and reservoir computing networks, to predict extreme events in dynamical systems. Their sensitivity analysis revealed that while feedforward networks were highly sensitive to noise and hyperparameters, LSTM networks showed greater robustness, and reservoir computing networks consistently delivered superior performance in predicting complex behaviors; LSTM networks are therefore worth investigating in this work. McCartney, Indlekofer, and Polifke [86] focused on precursor identification for thermoacoustic instabilities by applying supervised learning methods, where pressure signals were first categorized using Hidden Markov Models and then analyzed further after being transformed through detrended fluctuation analysis. Ruiz et al. [108] extended this approach by employing recurrence networks in combination with CNNs to classify the dynamic states in the pressure data, effectively differentiating between aperiodic and periodic structures indicative of instability onset. These approaches emphasize the importance of identifying dynamical structure in time series signals which is a principle that also underlies this thesis, where

unsupervised learning is used to uncover precursor states from latent representations of combustion dynamics.

Supervised learning has been very popular in cases ranging from combustion related studies, to purely data driven time series. However, supervised learning requires training data. When attempting to predict extreme events/precursors in chaotic time series, a label would be required for supervised learning, which is very difficult to obtain because of data scarcity. Therefore, unsupervised learning should be investigated as well.

4.2.2. Unsupervised Learning

Unsupervised learning is a fundamental approach in machine learning that enables models to discover patterns and structures within unlabeled data. Unlike supervised learning, where models learn from labeled examples, unsupervised learning algorithms operate independently to identify inherent relationships within datasets. The primary tasks of unsupervised learning include clustering, association rule mining, and dimensionality reduction. Clustering algorithms, such as K-means and hierarchical clustering, group data points based on similarities, making them useful for applications like customer segmentation and image analysis. Association rule learning, exemplified by the Apriori algorithm, identifies relationships between variables in large datasets, often employed in recommendation systems and market basket analysis. Dimensionality reduction techniques, including principal component analysis (PCA) and autoencoders, help manage high-dimensional data by preserving essential features while reducing complexity. These approaches allow for efficient data analysis, revealing hidden structures that can be utilized in various fields, from finance to healthcare [60].

Latent Representation Algorithm

Due to the large amount of data that the computational fluid dynamics (CFD) simulations produces, a latent representation algorithm is necessary to decrease the data size. Using an algorithm for this ensures that the important features of the data are well-represented, yet the computational load is much lower.

Jonnalagadda et al. [68] introduced a co-kurtosis PCA method that uses fourth-order statistical moments to capture stiff chemical dynamics more effectively than traditional PCA. The authors found that the co-kurtosis based reduced manifold not only better reconstructed the original thermo-chemical state but also provided more accurate predictions of species production and heat release rates, particularly in regions where rapid chemical reactions occur. This is relevant for latent space construction; co-kurtosis PCA may capture critical, extreme precursor states better than linear PCA. Malik et al. [83] applied advanced manifold learning techniques, including local PCA, to high-fidelity reacting flow simulations to reduce data dimensionality and enable unsupervised classification of different combustion regimes. The findings demonstrated that the approach significantly reduced the number of transport equations and system stiffness, while still preserving critical physical details, thus offering a computationally efficient way to analyze complex combustion data. Finally, Amaduzzi et al. [6] focused on quantifying parametric uncertainty in simulations of a hydrogen-fueled flameless combustion furnace by coupling dimensionality reduction with nonlinear regression. The study concluded that by isolating the dominant modes of the combustion process and constructing surrogate models, the method effectively captured key combustion parameters, such as temperature profiles and emission indices, while substantially reducing the computational cost.

Although both PCA and autoencoders are effective methods for reducing dimensionality, they have various uses and perform better in particular situations. When interpretability and processing speed are more important, PCA is a great option for straightforward, linearly separable data. However, complicated, non-linear data that needs a more adaptable model to capture complex patterns are better

suited for autoencoders. In this study, PCA is still a dependable, user-friendly method for rapid dimensionality reduction and data exploration, however, autoencoders may provide a more potent answer for non-linear data for many real-world tasks [61].

In a further study, Xu, Yang, and Zhang [132] introduce a novel deep learning framework that combines a bidirectional LSTM variational autoencoder with a Wasserstein distance–based classifier. They found that this approach successfully projects the high-dimensional spatio-temporal data from circular flame arrays into a low-dimensional latent space with non-overlapping clusters. This clear separation enables effective unsupervised recognition and classification of various oscillatory modes, outperforming traditional techniques like PCA and standard VAEs. Referring to this work, this is directly applicable, showing how latent spaces can separate combustion states, much like flashback precursor clustering. Furthermore, Ding et al. [33] present a convolutional autoencoder–based reduced order model to emulate the spatial distributions of key combustion variables. The study shows that by capturing the nonlinear features of the combustion process, the autoencoder-based reduced order model (ROM) achieves higher prediction accuracy than conventional POD-based models, while reducing computational time by several orders of magnitude. This is useful for understanding nonlinear feature extraction; a convolutional AE could be adapted for spatial CFD fields leading to flashback. In another study, Xu et al. [131] apply a multi-channel VAE to high-speed imaging data from a scramjet combustor. The method extracts critical temporal features and clusters the time-series data into distinct groups corresponding to different combustion modes. The results demonstrate that this unsupervised clustering approach reliably distinguishes between dynamic combustion states, providing a valuable tool for diagnosing and controlling scramjet combustion instabilities. This further reinforces that variational autoencoder based clustering works on combustion time series; relevant for this precursor state separation goal.

Iemura et al. [63] analyzed cool flame oscillations around a fuel droplet array using a variational autoencoder with proper orthogonal decomposition (VAE-POD) method that combines variational autoencoder–based nonlinear attractor reconstruction with mode decomposition. The study concluded that this novel approach successfully decomposed the complex spatiotemporal oscillations of multi-phase, multi-species reacting fluids, thereby clarifying the underlying physical mechanisms governing cool flame behavior. The output of interest for this study is the number of latent variables of the output of the VAE. These latent variables capture oscillatory behavior, which is relevant for identifying flashback precursor patterns. These can be visualized in the phase plane from [63], shown in Figure 4.1

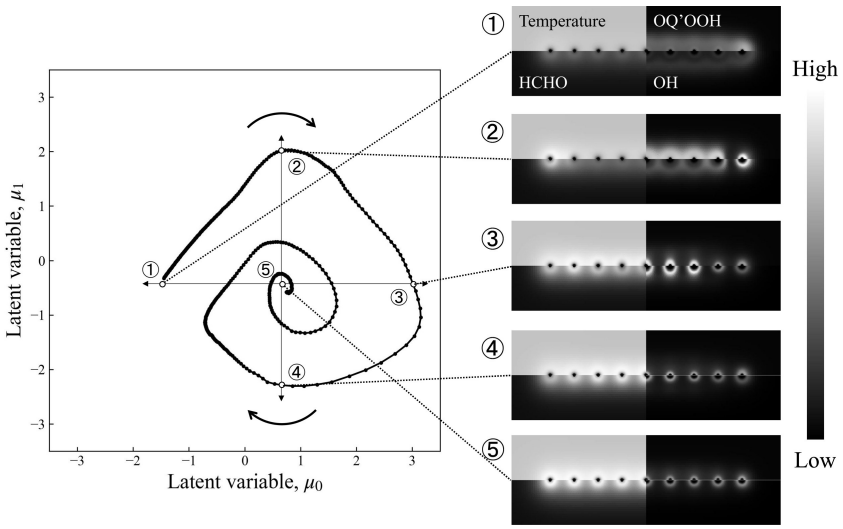


Figure 4.1: Latent Variable Phase Space Representation [63]

Precursor Detection Algorithm

A precursor algorithm to identify ignition kernels is essential to predict the onset of flashback.

For this purpose, studies involving clustering have been very popular. Fichera, Losenno, and Pagano [41] study focuses on the nonlinear dynamics of a lean gas turbine combustor. The authors collected pressure or flame emission signals under various operating conditions and then applied clustering techniques to the chaotic time series data. By grouping similar dynamical behaviors, the methodology helped to classify different combustion regimes, ranging from stable periodic operation to intermittent or fully chaotic behavior. This is extremely important to this work, as it is a similar combustion system and time series clustering goal; a useful baseline for flashback classification. Next, Han, Yang, and Song [52] addresses the challenge of predicting chaotic time series over multiple time steps. It uses a local Volterra model, a nonlinear system identification approach, to approximate the underlying dynamics. To enhance prediction accuracy and efficiency, the study proposes clustering the phase space points (obtained via time-delay embedding) to identify optimal neighboring points for model training. In essence, by grouping similar phase points, the method selects the most representative local data, reducing computational load while capturing the system's evolution. This clustering helps localize relevant dynamics before modeling, similar to the latent clustering performed in this work. In another study, Widiputra et al. [128] propose an evolving clustering framework that continuously adapts as new data arrives. The core idea is to extract and group local patterns (modeled by polynomial regression) from chaotic time series. The evolving nature of the algorithm allows it to capture repeating patterns that change over time, which is particularly useful for datasets such as stock prices or currency exchange rates. By associating each cluster with a characteristic polynomial function, the method not only predicts future values with high accuracy but also provides insights into the underlying structure of the chaotic process. This highlights the value of dynamic, adaptive clustering, which could be applied to changing precursor structures in CFD time series like this work.Finally, Kirichenko, Pichugina, and Zinchenko [72] take a feature-based approach to clustering time series that exhibit complex (often chaotic) behavior. Instead of comparing raw time series data point by point, the authors first extract a set of global statistical indicators, such as measures of trend, seasonality, autocorrelation, and even indicators linked to chaos, from each series. These feature vectors then serve as the basis for clustering (using algorithms like k-means). By doing so, the work demonstrates that even subtle differences in the dynamical regimes (for instance, slight changes in chaos intensity from logistic maps) can be effectively captured and grouped. This approach not only improves clustering accuracy but also reduces the dimensionality of the problem, making it scalable to larger datasets. It suggests that higher-level statistics (e.g. entropy, autocorrelation) could be used as inputs/features in the latent space model.

Specific studies in combustion have also been prominent. Ji et al. [66] focused on improving safety evaluations for flammable liquids used in compression ignition engines. In this study, the authors compiled a comprehensive database of liquid compounds, including not only conventional flammability parameters (like flash point) but also properties related to flame propagation and aerosol formation. Two unsupervised clustering methods, k□means and spectral clustering, were applied to group the compounds based on their combustion-related characteristics. The study used the global mean silhouette value to determine the optimal number of clusters and assess clustering performance. Spectral clustering was found to outperform k-means, resulting in a more accurate classification of risk ratings. In addition, the authors employed PCA and star coordinate diagrams to visualize the high-dimensional data in 2D, making the cluster structures more interpretable. Ultimately, the clustering results were integrated with an information entropy approach to develop a novel combustion risk index. Furthermore, Yu et al. [135] combined detailed CFD simulations with the K□means clustering algorithm to gain a localized understanding of soot formation in an engine combustion chamber under varied operating conditions. First, the CFD simulations produce high-resolution scalar distributions, primarily local equivalence ratio, and temperature, which are important parameters influencing soot generation. The K-means algorithm is

then applied to these datasets to automatically partition the combustion chamber into distinct zones, categorizing them into areas with relatively low and high soot deposition. By correlating these clusters with operating conditions, the study reveals how changes in parameters like oxygen concentration and combustion boundaries affect both the spatial distribution and magnitude of soot production. By using unsupervised clustering to analyze CFD outputs, it becomes similar to this work.

Finally, Golyska and Doan [46] investigated how data-driven clustering techniques can be employed to detect early-warning signals of extreme events in chaotic dynamical systems. In many natural and engineered processes, extreme events can have catastrophic consequences. The study proposes a modularity-based clustering framework that analyzes state-space data to identify precursor states that statistically lead to such extreme events. This framework involves partitioning the system's state space into clusters, constructing probability transition matrices between these clusters, and using state-space tessellation to delineate regions that serve as early indicators. The approach was validated on benchmark chaotic systems, including a turbulence model and a two-dimensional Kolmogorov flow, both of which exhibit bursts in kinetic energy and dissipation. The clustering algorithm successfully isolated the precursors, providing a probabilistic means to forecast the onset of extreme events.

Floris et al. [43] extends the previously established clustering framework, originally detailed in Golyska and Doan [46]'s paper to a real-world combustion application. In this study, the authors apply an unsupervised, data-driven clustering technique to sensor data from a lean hydrogen reheat combustor, with the goal of detecting early precursor states that signal the onset of flashback. By successfully identifying these precursor states, the article not only confirms the validity of the earlier clustering-based methodology but also demonstrates its practical efficacy in predicting flashback events.

4.3. Research Questions

For this study, the modularity based clustering algorithm for precursor identification, developed by Golyska and Doan [46] and further utilized by Floris et al. [43] will be employed. This data□driven technique has been proven effective in complex systems and is particularly well□suited for turbulent reacting systems, which are inherently chaotic and high-dimensional due to their wide range of spatio-temporal scales and perturbations.

Furthermore, data is obtained from LES simulations under multiple scenarios. To continue Floris [42]'s study and bring it one step closer to application, the sampling locations of the LES model will be in areas more accessible and non-intrusive compared to [42]. Moreover, to manage the computational load while preserving the essential nonlinear features of the chaotic combustion dynamics, an AE is employed similar to [63] to obtain a low-dimensional latent representation of the high-dimensional time series. The AE is adept at capturing the complex, nonlinear interactions in turbulent reacting flows by encoding the underlying patterns into a compact latent space. The latent variables extracted from the AE are then used as inputs to the modularity-based clustering algorithm, which partitions the latent space into distinct clusters corresponding to potential precursor states. This integrated approach enables the identification of early-warning signals for flashback events and enhances real-time monitoring by reducing the data dimensionality without losing critical dynamic information.

Finally, the secondary part of this study will focus on water injection to suppress flashback. Floris [42] had performed a similar study focusing on tuning the parameters of the water spray; however, he noted that the flashback would return. For this purpose, the timing and tapering of the spray will be given special attention and simulated in a LES. Based on the literature performed, the following research questions can be posed:

1. To what extent can the modularity-based clustering algorithm, when applied to the AE-produced latent variables, accurately identify precursor states to flashback events as simulated in a high-

pressure LES with practical sampling locations?

2. To what extent can water injection suppress a detected flashback?

5

Simulation

In this chapter the numerical configuration used to generate the data is described. The dry LES setup is detailed, including the governing equations, subgrid modelling, domain and mesh, boundary and operating conditions, and numerics. The rationale for choosing LES over RANS and DNS is discussed with reference to fidelity and computational cost, and validation checks (e.g., spectral content and resolved kinetic energy) are reported to demonstrate adequacy of resolution. The resulting flow and thermo-chemical fields are summarized to highlight the dominant unsteady mechanisms—autoignition, relaxation, and flashback—that are later used as targets for precursor detection.

5.1. Dry LES Configuration

Computational fluid dynamics (CFD) is a cornerstone of modern engineering, offering insights that are nearly impossible to obtain through physical experiments alone. From optimizing aircraft aerodynamics to improving combustion in engines or predicting blood flow in arteries, CFD enables engineers to visualize and quantify complex fluid behaviors with extraordinary detail. It saves time, reduces cost, and allows for exploration far beyond what traditional testing can offer.

Among the many tools in CFD, there are three primary approaches used to simulate turbulent flow, each offering a trade-off between accuracy and computational cost: Reynolds-Averaged Navier-Stokes (RANS), Large Eddy Simulation (LES), and Direct Numerical Simulation (DNS). Their differences are highlighted below.

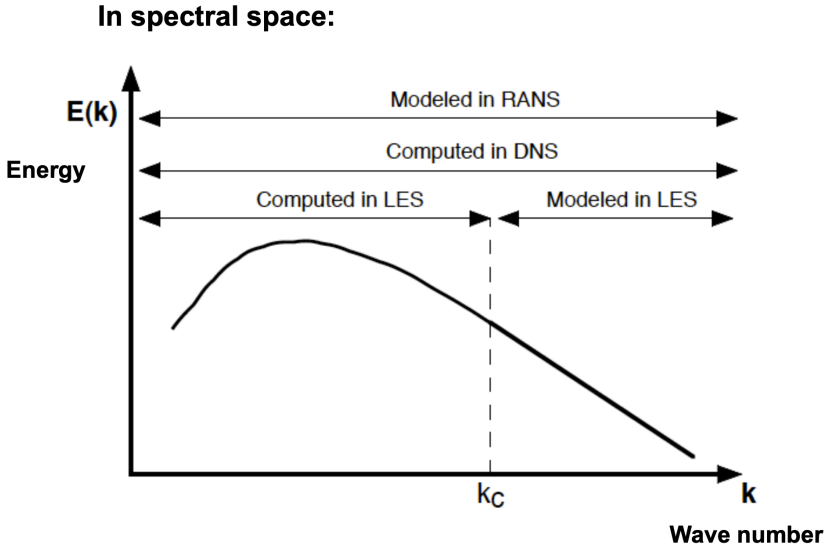


Figure 5.1: DNS, LES, and RANS comparison [93]

As shown in Figure 5.1, these methods differ in how they handle turbulence across scales. DNS resolves all turbulent eddies directly, capturing the full spectrum of flow behavior with the highest accuracy, but at immense computational cost. RANS, on the opposite end, models all turbulence, making it far more efficient but less detailed. LES strikes a middle ground: it resolves the large-scale turbulent structures explicitly while modeling the smaller, subgrid scales. This balance between fidelity and feasibility makes LES the method of choice for this work.

Since LES provides high-fidelity detail, it requires incorporating a wide range of physical phenomena to reach converged results. These may include heat transfer, chemical kinetics, and molecular diffusion. In more complex cases, the model must also capture two-phase flow dynamics, where interactions between liquid droplets, such as fuel or water, and the surrounding gas are significant. As a result, accurately simulating turbulent combustion calls for a comprehensive and integrated approach that accounts for all these coupled processes to reflect the system's true behavior.

5.1.1. Governing Equations

In a LES, the governing equations are filtered Navier–Stokes equations, which are derived from the Navier–Stokes equations by applying a spatial filter to separate large, resolved turbulent scales from small, subgrid scales. The CFD used in this work, Converge CFD, uses the following equations: The mass transport equation,

$$\frac{\delta \rho}{\delta t} + \frac{\delta \rho u_i}{\delta x_i} = S \tag{5.1}$$

and the momentum transport equation,

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + S_i \tag{5.2}$$

where σ_{ij} is the viscous stress tensor, and is given as:

$$\sigma_{ij} = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu_t \left(\frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \tag{5.3}$$

In these equations, ρ is density, u is velocity, P is pressure, δ_{ij} is the Kronecker delta, u_t is turbulent velocity, and S is a source term. Next, the energy equation is given as:

$$\frac{\partial \rho e}{\partial t} + \frac{\partial \rho e u_j}{\partial x_j} = -P\frac{\partial u_j}{\partial x_j} + \frac{\partial}{\partial x_j}\left(K_t \frac{\partial T}{\partial x_j}\right) + \sigma_{ij}\frac{\partial u_i}{\partial x_j} + \frac{\partial}{\partial x_j}\left(\rho \sum_m D_m h_m \frac{\partial Y_m}{\partial x_j}\right) + S \tag{5.4}$$

Here, e is the specific energy, K_t is turbulent conductivity, D_m is the species mass diffusion coefficient, h_m is the species specific enthalpy, and Y_m is the mass fraction of species m. The turbulent conductivity is given by:

$$K_t = K + c_p \frac{\mu_t}{Pr_t} \tag{5.5}$$

where K is the conductivity, c_p is the specific heat at constant pressure, and:

$$Pr_t = \frac{c_p \mu_t}{k_t} \tag{5.6}$$

is the turbulent Prandtl number. Finally, the species conservation equation is given by:

$$\frac{\partial Y_m \rho}{\partial t} + \frac{\partial Y_m \rho u_j}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\rho D_m \frac{\partial Y_m}{\partial x_j}\right) + S_m \tag{5.7}$$

where S_m is a general source term to account for evaporation, chemical reactions, and submodels, and D_m is the local mixture-averaged diffusion coefficient. It is calculated as:

$$D_m = \frac{1 - X_m}{\sum_{j, j \neq m} X_j \left(\frac{1}{D_{mj}}\right)} \tag{5.8}$$

where X_m is the mole fraction of species m, and D_{mj} is the binary diffusion coefficient for species m and j. In turbulent conditions, an extra term of turbulent mass diffusion coefficient $D_t = \frac{v_t}{Sc_t}$ must be added to D_m.

Once the equations are determined, the final LES equations can be obtained by resolving the large scale turbulence and modeling the subgrid scale. This can be done by filtering the relevant quantities Q. In this LES, a mass-weighted, a Favre filtering is used:

$$\bar{\rho}\tilde{Q}(\mathbf{x}) = \int \rho Q(\mathbf{x}^*) F(\mathbf{x} - \mathbf{x}^*)\, d\mathbf{x}^* \tag{5.9}$$

Here, F represents the LES filter, and the filtered variable $\tilde{Q}$ is defined as $\tilde{Q} = \overline{\rho Q}/\bar{\rho}$. In this context, F is implemented as a box filter, and the filter width Δ is related to the size of the computational cell. Specifically, Δ is given by $\Delta = \sqrt[3]{V}$, where V is the volume of the cell.

With this filtering approach, certain quantities remain unresolved and must be modeled. These include the subgrid-scale Reynolds stresses $\widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j$, which require a turbulence model. Also requiring modeling are the unresolved species transport terms $\widetilde{u_j Y_k} - \tilde{u}_j \tilde{Y}_k$ and enthalpy transport terms $\widetilde{u_j h_t} - \tilde{u}_j \tilde{h}_t$. In addition, the filtered laminar diffusion fluxes $\bar{J}_j^k$ and $\bar{J}_j^h$, as well as the filtered chemical reaction rate $\bar{\dot{\omega}}_k$, must be accounted for.

5.1.2. Subgrid Scale Modeling

Each subgrid scale quantity must be accounted for. For this work, the one-equation viscosity model is used to account for subgrid-scale Reynolds stresses. Originally developed by Yoshizawa and Horiuti [134] and refined by Menon, Yeung, and Kim [87], this model uses a subgrid-scale kinetic energy

equation to represent turbulent viscosity. It is shown as:

$$\frac{\partial k}{\partial t} + \bar{u}_i \frac{\partial k}{\partial x_i} = -\tau_{ij} \frac{\partial \bar{u}_i}{\partial x_j} - \epsilon + \frac{\partial}{\partial x_i} \left(\frac{\nu_t}{\sigma_k} \frac{\partial k}{\partial x_i} \right) \tag{5.10}$$

The equation includes terms on the right-hand side representing production, dissipation, and diffusion. Furthermore, the kinetic energy, turbulent viscosity, and the sub-grid dissipation are:

$$k = \frac{1}{2} \left(\overline{u_i u_i} - \bar{u}_i \bar{u}_i \right) \tag{5.11}$$

$$\nu_t = C_k \sqrt{k} \Delta \tag{5.12}$$

$$\epsilon = \frac{C_\epsilon k^{3/2}}{\Delta} \tag{5.13}$$

Finally, the subgrid scale stress tensor is:

$$\tau_{ij} = -2\nu_t \bar{S}_{ij} + \frac{2}{3} k \delta_{ij} \tag{5.14}$$

In these supporting equations, C_k is the viscosity constant, C_ϵ is the subgrid scale dissipation constant, and σ_k is the reciprocal subgrid scale kinetic energy Prandtl number. They are set to:

$$C_k = 0.005 \tag{5.15}$$

$$C_\epsilon = 1 \tag{5.16}$$

$$\sigma_k = 1 \tag{5.17}$$

as is recommended by the authors of the citations. Next, the chemical kinetics must be modeled.

5.1.3. Chemistry Model

To solve the chemical kinetics, the SAGE solver is used, which operates based on CHEMKIN input formats. The description that follows is based on SAGE as implemented in Converge and Senecal et al. [112]. Within Converge, the CVODE solver is employed to integrate the system of ordinary differential equations (ODEs). The SAGE model in Converge also supports third-body reactions, allowing species-specific efficiency factors. In addition, it can handle pressure-dependent reactions using the Lindemann, Troe, SRI, or PLOG formulations [104].

The SAGE chemistry model computes reaction rates for each elementary step, while the CFD solver subsequently solves the associated transport equations. A multi-step chemical reaction can be expressed as:

$$\sum_{k=1}^{N} \nu'_{k,i} f_k \leftrightharpoons \sum_{k=1}^{N} \nu''_{k,i} f_k \quad \text{for } i = 1, 2, \ldots, I \tag{5.18}$$

where $v'_{k,i}$ and $v''_{k,i}$denote the stoichiometric coefficients for the reactants and products, respectively, k represents the species, i the reaction index, I the total number of reactions, and f_k the chemical symbol for the corresponding species. The production rate of a given species is described by:

$$\dot{\omega}_k = \sum_{i=1}^{I} \nu_{k,i} q_i \quad \text{for } k = 1, 2, \ldots, N \tag{5.19}$$

for a total of N species, and $\nu_{k,i} = \nu_{k,i}^{''} - \nu_{k,i}^{'}$. The rate of progress parameter, q_i, is defined as:

$$q_i = k_{i,f} \prod_{k=1}^{N} [X_k]^{\nu_{k,i}^{'}} - k_{i,r} \prod_{k=1}^{N} [X_k]^{\nu_{k,i}^{''}} \tag{5.20}$$

where $[X_k]$ is the molar concentration of species k, and $k_{i,f}$ and $k_{i,r}$ and the forward and reverse rate coefficients for a reaction i. The forward rate coefficient can be found using the Arrhenius equation:

$$k_{i,f} = A_i T^{\beta_i} \exp\left(\frac{-E_i}{RT}\right) \tag{5.21}$$

where A_i is the pre-exponential factor, T is the temperature, β_i is the temperature exponent, E_i is the activation energy and R is the ideal gas constant. The reverse can be calculated by dividing Equation 5.21 by $K_{i,c}$ where

$$K_{i,c} = K_{i,p} \left(\frac{P_{\text{atm}}}{RT}\right)^{\sum_{m=1}^{M} \nu_{mi}} \tag{5.22}$$

and in turn,

$$K_{i,p} = \exp\left(\frac{\Delta S_i^0}{R} - \frac{\Delta H_i^0}{RT}\right) \tag{5.23}$$

and

$$\frac{\Delta S_i^0}{R} = \sum_{k=1}^{N} \nu_k \frac{S_k^0}{R} \tag{5.24}$$

$$\frac{\Delta H_i^0}{RT} = \sum_{k=1}^{N} \nu_k \frac{H_k^0}{RT} \tag{5.25}$$

where H and S are enthalpy and entropy, respectively. In a computational mesh, the governing equations can now be written according to the above equations. These are the mass,

$$\dot{\omega}_k = \frac{d[X_k]}{dt} \tag{5.26}$$

and energy equation

$$\frac{dT}{dt} = \frac{\sum_k \left(\bar{h}_k \, \dot{\omega}_k\right)}{\sum_k \left([X_k] \, \bar{c}_{p,k}\right)} \tag{5.27}$$

In the energy equation, $\dot{\omega}_k$ is computed based on Equation 5.19, while $\bar{h}_k$ and $\bar{c}_{p,k}$ represent the molar specific enthalpy and the molar specific heat at constant pressure, respectively. The heat release $\dot{Q}$, volume V, and species-specific thermodynamic properties such as h_k and $c_{p,k}$ (for each species k) are used to solve the temperature evolution at each time step. In Converge, the temperature that is calculated from the energy equation is used to update the forward and reverse reaction rate coefficients iteratively, repeating the process until convergence is achieved. Notably, this updated temperature is used for the chemistry calculations rather than the actual cell temperature, which is only adjusted after detailed chemistry has converged, using the resulting species concentrations. Furthermore, to ensure accuracy of the LES, a thickened flame model (TFM) will be utilized.

5.1.4. Thickened Flame Model

The TFM is a mesh refinement method that acts at the flame front, capturing better flame front dynamics. This is because the mesh is generally not fine enough to capture these dynamics, therefore the flame thickness increases without changing the laminar flame speed, which removes the need for subgrid scale models.

Firstly, the laminar flame thickness and speed must be found. This is accomplished by performing 1D simulations across a range of inlet conditions by varying temperature, equivalence ratio, and pressure. The TFM then interpolates these results. According to Turns [118], the laminar flame speed follows the relation $s_L^2 \propto \alpha\,\dot{\omega}/[\mathrm{H_2}]$, where α is the thermal diffusivity of the mixture, $\dot{\omega}$ is the reaction rate, and $[\mathrm{H_2}]$ is the fuel concentration. Furthermore, the thermal diffusivity scales with temperature and pressure according to the relation $\alpha \propto T_u \bar{T}^{3/2} P^{-1}$, where $\bar{T}$ is the mean temperature between the unburnt and burnt gases. Both the reaction rate and the fuel concentration can also be described as functions of temperature and pressure as follows:

$$\dot{\omega} \propto T_b^n P^m \exp\left(-\frac{E_A}{RT_b}\right) \qquad [F] \propto \frac{P}{T_u} \tag{5.28}$$

Substituting these relationships, the laminar flame speed s_L can be expressed in terms of the burnt and unburnt temperatures, as well as pressure. The flame thickness δ_l is then obtained from s_L using the relation $\delta_l = \alpha/2s_L$. This gives:

$$s_L \propto \bar{T}^{0.375} T_u T_b^{-n/2} P^{(n-2)/2} \exp\left(-\frac{E_A}{RT_b}\right) \tag{5.29}$$

$$\delta \propto \bar{T}^{0.375} T_b^{n/2} P^{-n/2} \exp\left(\frac{E_A}{RT_b}\right) \tag{5.30}$$

For this work, values for the laminar flame speed and flame thickness are adopted from the results computed by Floris [42]. These preliminary values highlight key trends. Notably, pressure has a dual effect: it reduces the flame speed at lower temperatures, but significantly increases it at higher temperatures due to enhanced hydrogen diffusion. This is particularly important in the context of flashback, as higher flame speeds increase the risk of flashback.

Moreover, increasing pressure also reduces the flame thickness. For instance, at 30 atm, δ_L becomes 20 times smaller compared to its value at 1 atm. It is worth noting that both flame speed and thickness are influenced by the chemical kinetics mechanism used. In this work, the mechanism developed by Li et al. [78] is used throughout, and no further modifications are necessary.

The thermal diffusivity itself scales as $\alpha \propto \bar{T}\,T^{3/2} P^{-1}$, where T is the average temperature between the unburnt and burnt gases. Both the reaction rate and the fuel concentration can also be expressed as functions of temperature and pressure, as described below.

Now that these parameters are determined, further explanation can be given on the TFM equations. In Converge, the TFM is based on the formulation by Legier, Poinsot, and Veynante [76], dynamically adjusts the flame structure and its interaction with turbulence using two key parameters: the thickening factor F and the efficiency factor E. The thickening factor multiplies the original laminar flame thickness, yielding $F \cdot \delta_L$, where F typically ranges from 10 to 100 in gas turbine applications. The efficiency factor accounts for subgrid scale flame wrinkling that is not directly resolved.

To maintain the correct flame dynamics despite artificial thickening, several physical properties are rescaled. The thermal diffusivity becomes $E \cdot F \cdot D$, the laminar flame speed is updated to $E \cdot s_L$, and the pre-exponential factor in the Arrhenius rate expression is modified to $\frac{E \cdot A}{F}$. These modifications enable direct resolution of the flame front without the need for filtering, similar to DNS, while leaving flow features far from the flame unaffected. As a result, the scalar conservation equation is reformulated

accordingly.

$$\frac{\partial \rho Y_i}{\partial t} + \frac{\partial \rho Y_i u_j}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\rho \cdot E \cdot F \cdot D \frac{\partial Y_m}{\partial x_j}\right) + \frac{E}{F}\dot{\omega}_i \tag{5.31}$$

where,

$$F = 1 + (F_{\text{max}} - 1)S \tag{5.32}$$

Here, S is the local flame sensor that determines the locality of the flame thickness, and F_{max} is

$$F_{\text{max}} = \frac{n_{\text{res}} \Delta x}{\delta_l} \tag{5.33}$$

For F_{max}, n_{res} is the number of grid points across the flame and is equal to 5. Δ_x is the local grid spacing. Furthermore, S can be calculated as

$$S = \max\left[\min\left(\beta\left(\frac{|\tilde{\omega}_{\text{sens}}|}{\bar{\Omega}_{\text{sens},0}(\phi)} - 1\right), 1\right), 0\right] \tag{5.34}$$

where $|\tilde{\omega}_{\text{sens}}|$ is the local reaction rate, $\bar{\Omega}_{\text{sens},0}$ is the maximum reaction rate of the sensor from a laminar flame at a given equivalence ratio, and β is a modeling parameter that determines the sensor thickness. Next, to improve the accuracy of the sensor, a self-adjusting sensor by Schulz et al. [111] is applied to the extremities of the flame to better capture the species gradients. It does so by utilized a passive indicator function, ψ, and is transported as,

$$\frac{\partial \bar{\rho}\tilde{\psi}}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_j\tilde{\psi}}{\partial x_i} = \frac{\partial}{\partial x_i}\left(F \Xi_\Delta \bar{\rho} \tilde{D}_\psi \frac{\partial \tilde{\psi}}{\partial x_i}\right) + \frac{\Xi_\Delta}{F}\bar{\omega}_\psi \tag{5.35}$$

where $\bar{\dot{\omega}}_\psi$ is a relaxation source term which varies predominantly based on τ_0, the local relaxation time, and $\tau_1 = \alpha\tau_c$. Here, τ_c is the characteristic flame time, defined as $\frac{\delta_L}{S_L}$, and α is a parameter that changes depending on whether the filtering is done upstream or downstream of the flame, due to the high temperature gradient. Therefore, $\bar{\dot{\omega}}_\psi$ is found as

$$\bar{\omega}_\psi = \begin{cases} -\frac{\tilde{\psi}}{\tau_1} & \text{if } S < 0.05 \\ \frac{\psi_0 - \tilde{\psi}}{\tau_0} & \text{if } S > 0.8 \\ 0 & \text{if } 0.8 > S > 0.05 \end{cases} \qquad \alpha_1 = \begin{cases} \alpha_{1,\text{ cold}} & \text{if } T \leq T_s \\ \alpha_{1,\text{ hot}} & \text{if } T > T_s \end{cases} \tag{5.36}$$

where T_s is the switch temperature. Finally, Ξ, the subgrid scale wrinkling factor remains as a parameter to be found. This parameter is used to account for the subgrid scale flame surface that may be lost when using a TFM, is defined as the ratio of the total flame surface to the resolved flame surface. Two models may be used to find Ξ. The first, developed by Charlette, Meneveau, and Veynante [25], assumes each turbulence function acts independently at the flame front:

$$\Xi_\Delta = \left(1 + \min\left[\frac{\Delta}{\delta_l} - 1,\ \Gamma_\Delta\left(\frac{\Delta}{\delta_l}, \frac{u'_\Delta}{s_l}, Re_\Delta\right)\frac{u'_\Delta}{s_l}\right]\right)^\beta \tag{5.37}$$

Γ_Δ is an efficiency function that accounts for the straining effects of all turbulence scales smaller than Δ. Here, Re_Δ represents the subgrid scale Reynolds number, and u'_Δ denotes the local turbulent velocity fluctuations. However, this model assumes that $\Delta/\sigma_L^0 \gg 1$, a condition that may not hold on finer computational grids. An alternative approach was proposed by Colin et al. [28].

$$\Xi_\Delta = 1 + \beta_{\text{Colin}} \frac{2\ln(2)}{3c_m + 5\left[\text{Re}_t^{1/2} - 1\right]} \Gamma_{\text{Colin}}\left(\frac{\Delta}{\delta_l}, \frac{u'_\Delta}{s_l}\right)\frac{u'_\Delta}{s_l} \tag{5.38}$$

where c_{ms} and β_{Colin} are parameters. Finally, the sensor S and the efficiency factor E can be rewritten:

$$S = \max\left[\min\left(\tilde{\psi}, 1\right), S\right] \tag{5.39}$$

$$E = \frac{\Xi|_{\delta=\delta_L}}{\Xi|_{\delta=F\delta_L}} \tag{5.40}$$

Next, the mesh will be configured.

5.1.5. Mesh

Creating a high-quality mesh has a significant impact on both the accuracy of the simulation results and the time required for convergence. A coarse mesh may lead to faster computation but at the cost of reduced accuracy. Conversely, an overly fine mesh can dramatically increase computational time without offering substantial improvements over a reasonably refined mesh. To understand the resolution required, the Pope criterion will be used [96], which states that at least 80% of the kinetic energy should be resolved by the mesh, as indicated below,

$$M(x,t) = \frac{k_r(x,t)}{K(x,t) + k_r(x,t)} \tag{5.41}$$

Here, K represents the turbulent kinetic energy of the resolved scales, and k_r denotes the subgrid scale turbulent kinetic energy. The variable M serves as a metric for the turbulence resolution. All three quantities are functions of both space and time. For reliable accuracy, the value of M is typically required to remain below 0.2 [96].

The computational grid is initialized with a baseline mesh size of 0.4 mm, which is considerably larger than the flame thickness under 20 atm conditions. To capture finer-scale phenomena, the simulation employs level 3 mesh embedding near walls and utilizes Automatic Mesh Refinement (AMR) based on subgrid scale variations in velocity and temperature. The maximum refinement level, denoted by s, is limited to 3, resulting in a refined mesh size of $\Delta x_{\text{new}} = \Delta x_{\text{base}}/2^s$. The subgrid scalar field, ϕ', is computed by subtracting the resolved scalar field $\bar{\phi}$ from the total scalar field ϕ, using a second derivative approximation derived from a Taylor series expansion [95],

$$\phi' \cong -\alpha_{[k]} \frac{\partial^2 \bar{\phi}}{\partial x_k \partial x_k} \tag{5.42}$$

This method, first used for temperature and other scalar fields, can be extended to vector fields as well. The mesh is refined when calculated values surpass specified thresholds, as long as the total cell count stays within a set limit (e.g., 10 million cells to maintain computational efficiency). If the values drop below 20% of the threshold, the mesh is coarsened. Finally, the TFM model also modifies the mesh by enforcing a minimum of 5 cells across the flame, ensuring refined resolution near the flame position. A visualisation of the mesh at an arbitrary timestep is shown below,

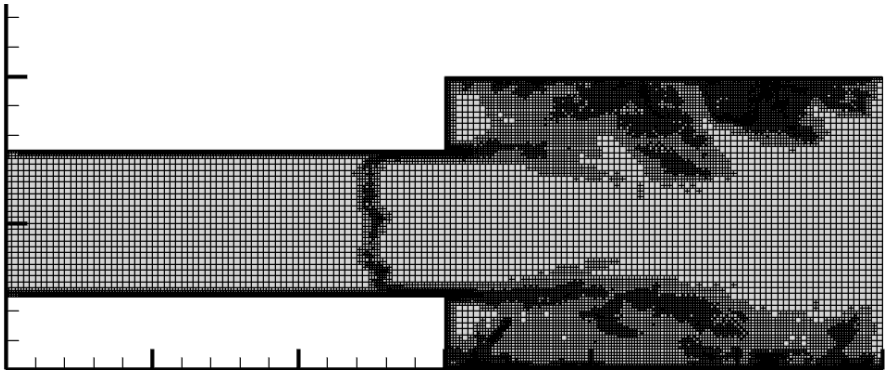


Figure 5.2: Mesh Setup with AMR

Once the mesh is defined in an adequate way, the numerical solver can be determined.

5.1.6. Numerical Methods

The numerical solution in Converge is based on the finite volume method (FVM) with implicit first-order time integration. Pressure–velocity coupling is achieved using the PISO algorithm [64], with Rhie–Chow interpolation [101] to prevent decoupling. Spatial discretization employs a hybrid central/upwind scheme with limiters to ensure stability, while turbulence variables use fully upwind differencing. The resulting linear systems are solved iteratively using Successive Over-Relaxation (SOR). A detailed derivation of the discretization, PISO pressure–velocity correction steps, Rhie-Chow interpolation, and SOR solver formulation is provided in Appendix A.

5.1.7. Boundary Conditions

Once the simulation technicals are determined, the boundary conditions can be set to start the simulation. For this work, a simplified version of the Ansaldo Energia GT36 engine is used. Specifically, the reheat combustor is of interest, which is displayed in 2D in Figure 5.3.

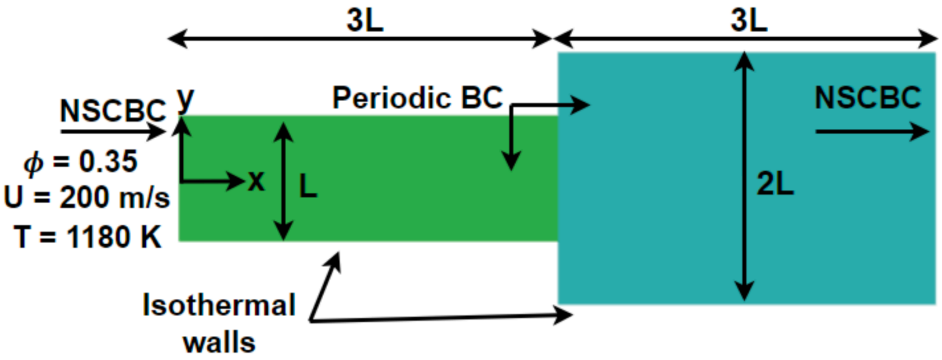


Figure 5.3: 2D Slice of the Ansaldo Energia GT36 - Simulated Geometry

This geometry is adapted from the work of Floris [42], Aditya et al. [4], and Rouco Pousada et al. [104]. The geometry consists of a mixing duct with dimensions 3L x L x 1.5L where L = 1 cm, and the combustion chamber of 3L x 2L x 1.5L. Furthermore, the general boundary conditions imposed at the inlet, walls, and outlet are listed below.

Inlet

The inlet conditions are imposed at the left-most of the mixing duct when observing Figure 5.3. Aditya et al. [4] uses a case where the inlet conditions contain the products of the first combustion stage; water, and a mixture of air and hydrogen. Furthermore, the velocity is uniformly set to 200 m/s while the equivalence ratio is at a constant $\phi = 0.35$. The pressure is set as a Neumann boundary condition, while the mass fraction is set to the values listed in Table 5.1 Next, the temperature must be determined.

Table 5.1: Species Mass Fraction at Inlet

Species	H_2	O_2	N_2	He	H_2O	Ar
Mass Fraction	0.007855	0.1780	0.7496	6.94E-07	0.05162	0.01286

A set 1100 K was used in the case of Aditya et al. [4], however Rouco Pousada et al. [104] indicates other parameters with reference to the dependence of the autoignition delay time with the temperature and pressure. Therefore, in this work, a delay time of approximately 0.15 ms is used with an inlet temperature of 1180 K in order to keep the flame at its design location.

A Navier-Stokes Characteristic Boundary Condition (NSCBC) is employed to minimize the reflection of acoustic disturbances at domain boundaries. This method is expanded on in Appendix B.

Walls

At all walls, a no-slip and isothermal boundary condition is imposed, with the wall temperature fixed at 750 K. Surface roughness effects are neglected, and subgrid scale turbulence near the wall is modeled using the law of the wall approach. In the z-direction of Figure 5.3, translationally periodic boundary conditions are applied. This configuration introduces a statistically homogeneous direction, which facilitates the collection of flow statistics and contributes to reduced computational cost.

Outlet

At the outlet, which is the far-right wall of the combustion chamber with respect to Figure 5.3, a simple physical boundary condition of Dirichlet is employed, with a pressure of 20 atm. Further, the velocity is a Neumann condition where backflow is neglected. Finally, $\sigma = 0.25$ similar to the inflow condition, where NSCBC is used.

5.1.8. Initial Conditions

To establish a steady state, and to summarize subsection 5.1.7, the computational domain is initialized with a uniform velocity of $200\,\mathrm{m/s}$ and a temperature of $1180\,\mathrm{K}$. The initial species composition matches the inflow, with the equivalence ratio set to $\phi = 0.35$, and the pressure fixed at $20\,\mathrm{atm}$. Following the procedure in the work of Floris [42] and Rouco Pousada et al. [104], the flow is first advanced for $1\,\mathrm{ms}$ under non-reacting conditions; combustion is then activated and the equivalence ratio is ramped gradually over the next $0.8\,\mathrm{ms}$, as proposed in the works of Kruljevic et al. [73] and Gruber et al. [50].

Once these are set, the dry simulation setup can be considered concluded, and the results are obtained. Next, the wet simulation will be outlined.

5.2. Wet Simulation

For the final simulation of this work, a wet simulation will be run to verify the precursors found. To do so, the parameters of the sprays must be configured. However, this can be a difficult task due to the injection process being quite complex, as it is a two-phase flow. Furthermore, due to the different stages of injection, as is shown in Figure 5.4.

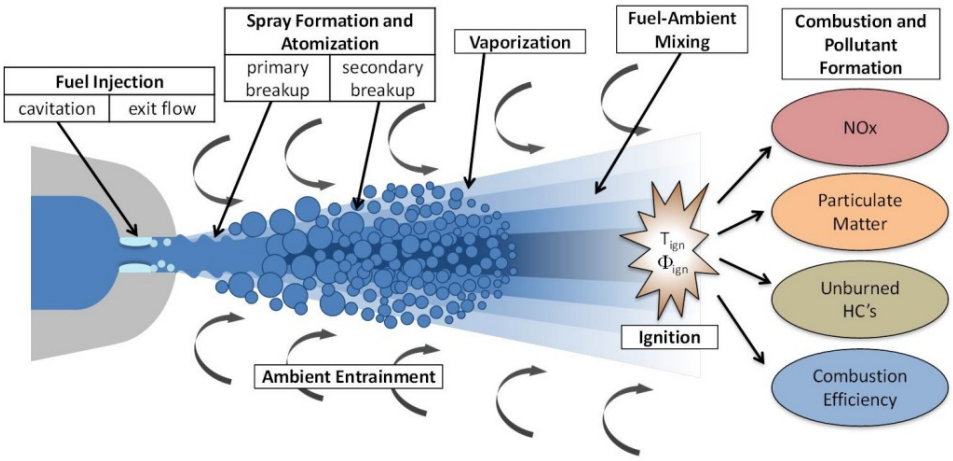


Figure 5.4: Water Injection Process [127]

The injection process begins as water enters the nozzle. The high flow velocity can reduce static pressure, leading to the formation of vapor cavities in the flow, which in turn affect how the spray develops. Upon entering the combustion chamber, the liquid undergoes primary breakup, where gas–liquid instabilities fragment the stream into droplets. This stage largely determines the subsequent spray behavior. The droplets then experience secondary breakup, in which aerodynamic forces break them into smaller fragments. At this point, droplets may either collide and merge into larger ones (coalescence) or further disperse into finer droplets. This cycle of coalescence and dispersion continues until the droplets reach the flame front and combust. Throughout this process, droplets are also subject to evaporation due to pressure differences and turbulent mixing. The complexity of these interacting mechanisms underscores the critical role of spray design parameters in injection systems.

An Euler-Lagrange modeling approach will be using in this work, where the gas phase is simulated using the Eulerian method, solving the flow properties in control volumes. The liquid phase is a Lagrangian method, which will track the droplet's motion relative to the local environment, including any interventions such as other collisions. The continuous gas phase and discrete droplets influence each other, making this approach possible. The droplet physics are implemented according to Converge [31], where breakup and coalescence are ignored. Details of the spray solver are listed in Appendix C, based on the report of Kruljevic et al. [73].

5.2.1. Spray design

To identify a spray configuration that suppresses flashback while respecting practical constraints, a concise set of tunable variables: atomizer type, injection location, nozzle diameter d_0, liquid mass flow rate $\dot{m}_L$, Sauter mean diameter (SMD), cone angle β, and cone thickness angle τ are used. These variables jointly determine the injection velocity V_{inj}, which is treated as an additional key design parameter. Swirl is deliberately excluded to avoid confined-jet vortex breakdown and to isolate autoignition-driven flashback.

As a suppression of the flashback is necessary, the work of Floris [42] is referenced. Since a very similar work and identical chamber is used in the simulation, the most successful configuration in that work will be used in this work. Therefore, unlike the exploratory approach, this work fixes each attribute to the best-performing values identified in [42] and report those choices alongside the rationale.

In relation to the reference work, a hollow-cone spray is used at the inlet with six nozzles to accelerate coverage of the mixing duct. The symmetric layout above is retained to maximize early-area coverage at fixed response time. This is shown in Figure 5.5

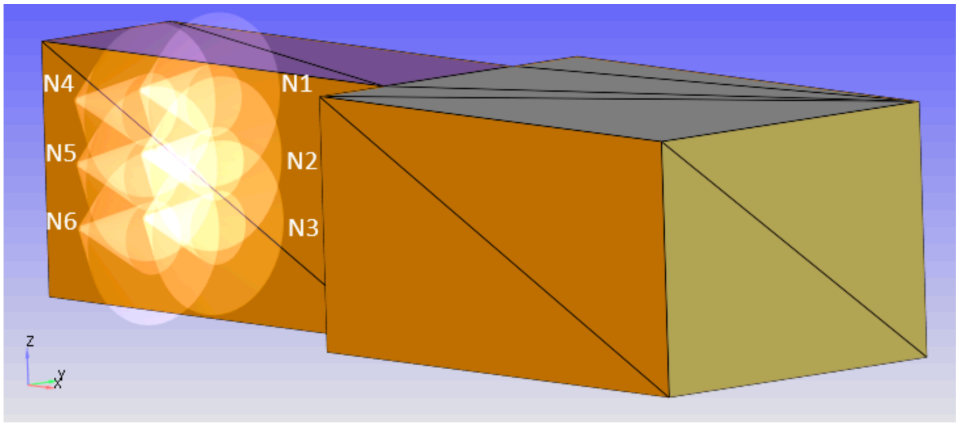


Figure 5.5: Nozzle Configuration

where,

Table 5.2: Nozzle Coordinates

Nozzle	N1	N2	N3	N4	N5	N6
x [cm]	0	0	0	0	0	0
y [cm]	0.25	0.25	0.25	-0.25	-0.25	-0.25
z [cm]	0.375	0	-0.375	0.375	0	0.375

Further, the other characteristics can be determined.

Geometry

These variables were explored empirically in the work of Floris [42], and the values that were found are in Table 5.3, where d_0 is the nozzle diameter. The angles are shown in Figure 5.6.

Table 5.3: Geometry Parameters

Parameter	Value	Units
d_0	0.1	mm
β	55	deg
τ	20	deg

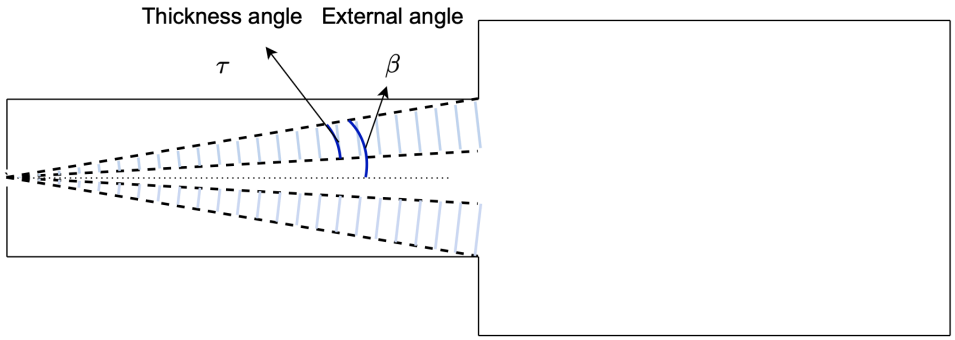


Figure 5.6: Spray Angle Definition

Here, the external angle β and thickness angle τ set the effective travel path.

Injection velocity

Injection velocity is a key design parameter for flashback suppression, as it directly influences the spray response time. For the selected inlet hollow-cone configuration ($x = 0$, $\beta = 55°$, $\tau = 20°$), the response velocity is estimated from

$$V_{\text{response}} = \frac{1}{\cos(\beta - \tau)} \frac{\Delta x}{t_{\text{pred}}}, \tag{5.43}$$

where Δx is the travel distance to the sampling location and t_{pred} the prediction time. The actual injection velocity is given by

$$V_{\text{inj}} = \frac{4\dot{m}_L}{\pi d_0^2 \rho_L}. \tag{5.44}$$

In this work, the target response velocity is $V_{\text{response}} \approx 610\,\text{m/s}$, and the injector geometry and mass flow rate are chosen such that $V_{\text{inj}} \approx 767\,\text{m/s}$, providing a margin above the requirement. Furthermore, the mass flow can be determined as the only remaining unknown variable, and is $\dot{m}_L$=0.006 kg/s per nozzle).

SMD

The SMD not only affects particle drag but also strongly influences droplet evaporation dynamics. It governs the timescale over which liquid droplets vaporize and, in doing so, absorb heat from the surrounding flow, producing a cooling effect. In water sprays not specifically designed for flashback suppression, the SMD must remain below a critical threshold; exceeding this limit results in droplets that cannot fully evaporate within the available residence time, thereby reducing overall spray efficiency. As used in Floris [42], SMD = $2e-5$ meters.

Atomization Regime Constraint

In this work, the Rosin-Rammler particle distribution is applied while neglecting break-up phenomena. Therefore, it is necessary to verify whether the spray's atomization regime aligns with this assumption. To do so, the criteria outlined by Floris [42], following the work of Reitz [99] are used. Reitz [99] explains that there are four atomization regimes, which occur sequentially as the injection velocity increases. In the Rayleigh Jet Breakup Regime, droplet formation is driven by axisymmetric surface oscillations caused by surface tension, resulting in droplet diameters larger than the jet itself. In the First Wind-Induced Breakup Regime, surface tension is supplemented by static pressure differences between the jet and the surrounding gas flow, causing breakup a few jet diameters downstream of the nozzle and producing droplets approximately the size of the jet diameter. The Second Wind-Induced Breakup Regime occurs when increased relative velocity between the liquid and gas phases amplifies short-wavelength surface waves, leading to droplet formation. Finally, in the Atomization Regime, fine droplets form immediately as the liquid exits the nozzle, fully atomizing the jet. These four regimes are illustrated in Figure 5.7.

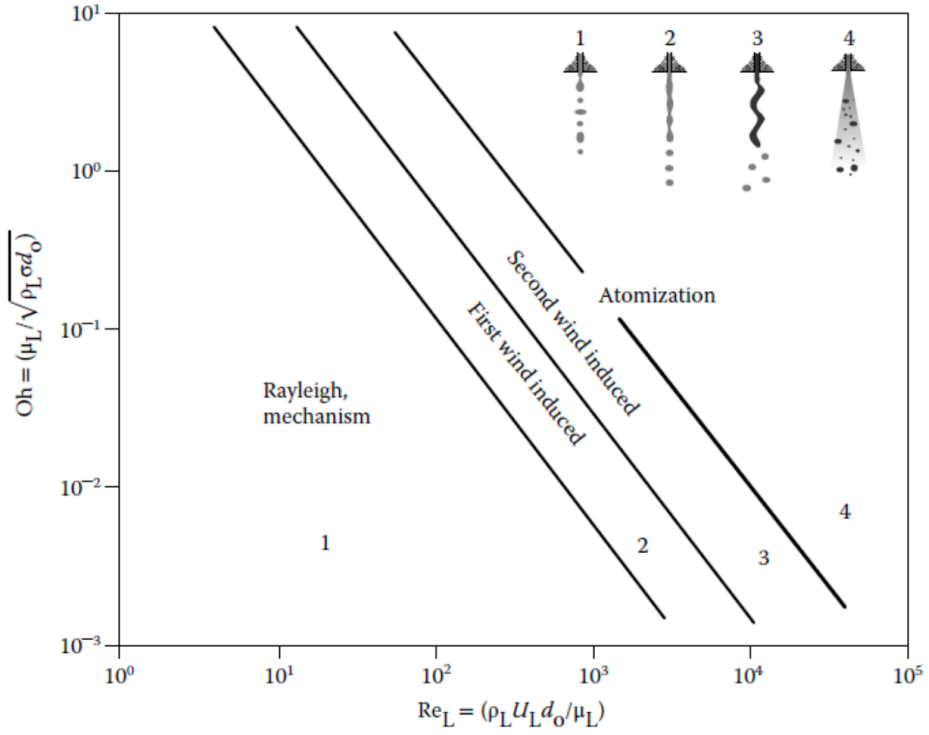


Figure 5.7: Regimes based on the Ohnesorge and Reynolds number [99]

To fit the constraint of the atomization regime, the design of the spray must be set so that the Reynolds and Ohnesorge numbers fit,

$$\mathrm{Re}_L = \rho_L V_{\mathrm{inj}} d_0 / \mu_L \tag{5.45}$$

and

$$\mathrm{Oh} = \mu_L / \sqrt{\rho_L \sigma d_0} \tag{5.46}$$

This is satisfied using the value of Floris [42], $\mathrm{Re}_L \approx 0.92 \times 10^5$ and $\mathrm{Oh} \approx 9.8 \times 10^{-3}$.

Expected Results

According to Floris [42], the adopted configuration achieves $\eta_e \approx 96.3\%$, indicating near-complete evaporation within the control volume of interest.

Table 5.4: Final Spray Parameters

Parameter	Value	Units
d_0	0.1	mm
$\dot{m}_L$ (per nozzle)	0.006	$\mathrm{kg/s}$
SMD	2e-5	m
β	55	deg
τ	20	deg
V_{response}	610	$\mathrm{m/s}$
V_{inj}	767	$\mathrm{m/s}$
Re_L	0.92e5	-
Oh	9.8e-3	-

Finally, the simulation sampling locations can be discussed.

5.2.2. Obtaining Data

Previously, it was noted that this study builds on the work of Floris [42], which focused on extracting data from the flame front. In contrast, the present work seeks to move the topic closer to practical application by obtaining data from more relevant regions of the simulation, such as the walls. Due to the nature of the flashback, the sampling locations are purely within the mixing duct, and is visible in Figure 5.8.

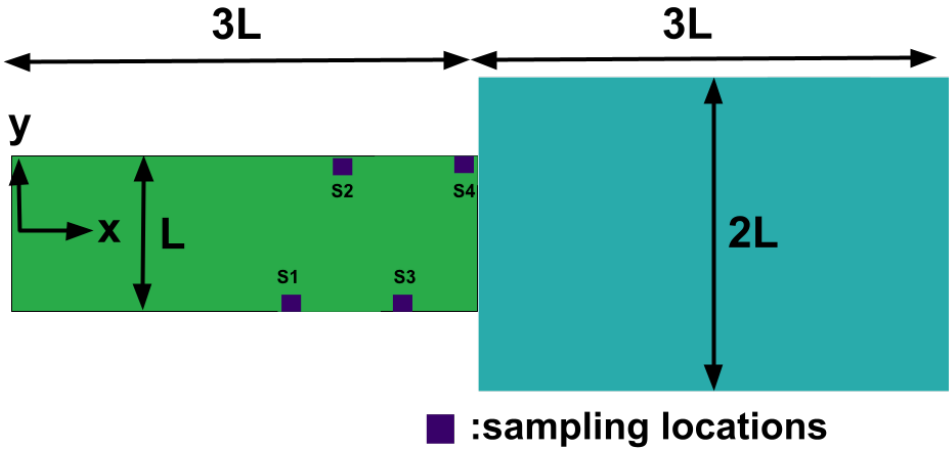


Figure 5.8: Sampling Locations

Four sampling points were selected at different heights and distances from the mixing duct outlet. This setup allows each point to be observed and evaluated to determine which provides the most reliable results.

6

Algorithm

This chapter specifies the end-to-end precursor-detection pipeline. The feature set extracted from the LES is introduced, after which dimensionality reduction is performed using an autoencoder to obtain a compact latent representation. The latent variables are segmented by clustering to define normal, precursor, and extreme regimes, and decision rules are stated for online mapping of unseen points to fixed clusters. Hyperparameters and regularization choices are documented, and the evaluation precursor time, false positives/negatives, and robustness tests across latent dimensionality are defined.

6.1. Dimensionality Reduction

The Large Eddy Simulation (LES) performed in this study produces a comprehensive set of physically relevant flow variables. As outlined in chapter 5, several distinct quantities are available, each providing insight into the underlying turbulent flow dynamics. While the full set of variables offers a detailed description of the system, directly supplying many features to the clustering algorithm is neither computationally efficient nor methodologically optimal. High-dimensional datasets are susceptible to the disadvantage of dimensionality, which can degrade clustering performance by obscuring intrinsic relationships between data points.

To mitigate this issue, a dimensionality reduction step is introduced. The primary objective is twofold: to reduce computational cost by lowering the dimensionality of the dataset, and to perform feature extraction, retaining only the most concrete structures and patterns relevant to the problem.

Autoencoders have demonstrated strong capability in this regard. As shown by Iemura et al. [63], autoencoder architectures can successfully compress high-dimensional flow data into a latent representation that is both compact and physically interpretable. The resulting latent variables can be readily projected into a phase space diagram, thereby enabling clearer visualisation of the system's dynamical behaviour.

Following the approach in [63], this work employs a similar autoencoder-based dimensionality reduction framework. The implementation utilises the Keras API from the TensorFlow library, chosen for its flexibility in model architecture design, efficient GPU utilisation, and integration with established deep learning workflows. The architecture will be discussed next.

6.1.1. Autoencoder Architecture

For an autoencoder, the architecture can be visualised in Figure 6.1.

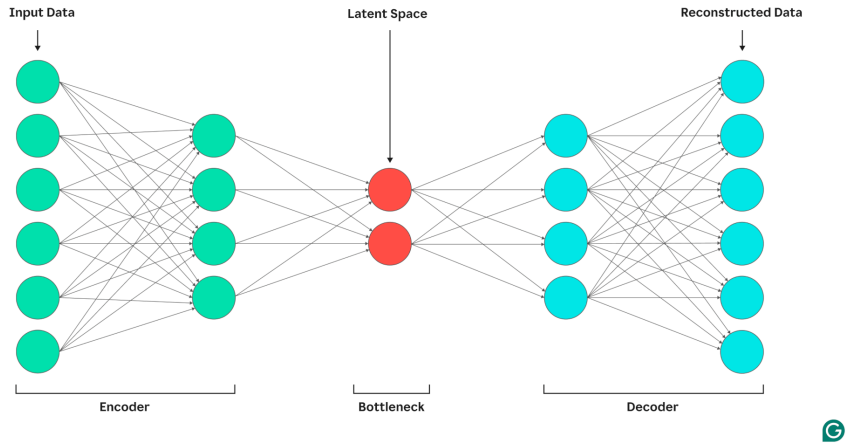


Figure 6.1: Autoencoder Architecture [48]

The schematic in Figure 6.1 illustrates a fully connected feedforward autoencoder. The architecture consists of three main components: the encoder, the bottleneck (latent space), and the decoder. The encoder, shown on the left, maps the high-dimensional input data onto a lower-dimensional representation through a series of hidden layers. Each circle denotes a neuron, and each directed edge represents a trainable weight parameter connecting two neurons in adjacent layers. The magnitude and sign of these weights determine the influence of one neuron on another in the forward pass.

The bottleneck layer, located at the center of the architecture, contains the smallest number of neurons in the network. This layer constitutes the latent space in which the compressed representation of the input data is stored. By constraining the capacity of this layer, the network is forced to learn an efficient encoding that preserves the most effective features of the original data while discarding redundancies, known as feature extraction.

Following the bottleneck, the decoder reconstructs the input data from the latent representation. The decoder mirrors the encoder in structure, progressively increasing the dimensionality of the data until it matches the original input space. As in the encoder, neurons in each layer are fully connected to neurons in the adjacent layer via trainable weights.

Each neuron in the network computes its output as a weighted sum of the outputs from the previous layer, to which a bias term is added. This sum is then passed through a non-linear activation function to introduce non-linearity into the model, enabling it to capture complex relationships in the data. Mathematically, the output y_j of neuron j in a given layer is expressed as

$$y_j = \phi\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right), \tag{6.1}$$

where x_i denotes the output of neuron i from the previous layer, w_{ij} is the weight connecting neuron i to neuron j, b_j is the bias associated with neuron j. The trainable parameters w_{ij} and b_j are optimised during training via backpropagation to minimise the reconstruction loss. The activation function, $\phi(\cdot)$, will be discussed next.

6.1.2. Activation Function

The activation function $\phi(\cdot)$ governs how the weighted sum of inputs to a neuron is transformed before being passed to the next layer. Its choice directly influences the representational capacity of the

network, the learning dynamics, and the interpretability of the latent space. Non-linear activation functions enable the network to model complex, non-linear relationships in the data, while linear activations preserve strictly linear transformations, which can be advantageous in cases where the underlying mapping is approximately linear or when interpretability is prioritised. Consequently, the selection of an activation function should be guided by the characteristics of the data, the intended use of the latent representation, and the reconstruction requirements. Several activation functions are commonly employed in autoencoder architectures:

The sigmoid activation maps any real-valued input into the interval $(0, 1)$, making it suitable for modelling probabilities or normalised features. It is defined as

$$\phi_{\text{sigmoid}}(z) = \frac{1}{1 + e^{-z}}. \tag{6.2}$$

While the sigmoid is smooth and differentiable, it suffers from the vanishing gradient problem for large positive or negative inputs, which can slow convergence in deep networks.

The rectified linear unit (ReLU) activation is defined as

$$\phi_{\text{ReLU}}(z) = \max(0, z). \tag{6.3}$$

It is computationally efficient and mitigates the vanishing gradient problem by allowing gradients to pass through unchanged for positive inputs. However, it can lead to “dead neurons” when weights are updated such that the neuron output remains at zero.

The linear activation function simply outputs the input without modification:

$$\phi_{\text{linear}}(z) = z. \tag{6.4}$$

It is typically used in the output layer of autoencoders when the reconstruction target contains continuous, unbounded values, allowing the network to produce unrestricted real-valued outputs.

In practice, different layers within an autoencoder may employ different activation functions. Non-linear functions such as ReLU or sigmoid are commonly used in the encoder and decoder hidden layers to capture complex dependencies in the data, while a linear activation is often adopted in the final output layer to facilitate accurate reconstruction of continuous-valued inputs. Furthermore, these facilitate backpropagation, which relies on a loss metric.

6.1.3. Loss Function

The loss function quantifies the discrepancy between the autoencoder’s reconstructed output and the original input, guiding the optimisation process during training. Its choice depends on the nature of the data, the scale of the variables, and the intended emphasis on specific reconstruction characteristics. In general, the objective is to minimise a measure of reconstruction error, ensuring that the latent representation retains the most relevant information for accurate reconstruction.

For continuous-valued data, the most widely used metric is the mean squared error (MSE), defined as

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x}_i)^2 , \tag{6.5}$$

where x_i and $\hat{x}_i$ denote the original and reconstructed values, respectively, and N is the total number of data points. MSE penalises larger errors more heavily, making it sensitive to outliers, and is suitable when reconstruction fidelity in terms of absolute magnitude is important.

In cases where different features or variables carry differing levels of physical importance or have varying scales, a weighted MSE can be used:

$$\mathcal{L}_{\mathrm{WMSE}} = \frac{1}{N}\sum_{i=1}^{N} w_i \left(x_i - \hat{x}_i\right)^2 , \tag{6.6}$$

where w_i is the non-negative weight assigned to the i-th feature or sample. By appropriately selecting w_i, the loss function can prioritise accurate reconstruction of variables deemed more significant for the application.

An alternative is the mean absolute error (MAE), given by

$$\mathcal{L}_{\mathrm{MAE}} = \frac{1}{N}\sum_{i=1}^{N} \left|x_i - \hat{x}_i\right| , \tag{6.7}$$

which applies a uniform penalty to all deviations, making it more robust to outliers and potentially better suited for data with heavy-tailed distributions.

For inputs that are normalised to the range $(0, 1)$ and can be interpreted probabilistically, the binary cross-entropy (BCE) loss is often employed:

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{N}\sum_{i=1}^{N} \left[x_i \log \hat{x}_i + (1 - x_i)\log(1 - \hat{x}_i)\right] . \tag{6.8}$$

This metric is particularly suitable for binary or probabilistic data, as it aligns with the probabilistic interpretation of certain activation functions, such as the sigmoid. In the present work, the reconstruction error is quantified using a MSE, due to all features being given equal weight.

6.1.4. Optimizer

The optimizer governs how the network's trainable parameters are updated during training to minimise the chosen loss function. It determines both the direction and magnitude of parameter updates based on the gradient of the loss with respect to the weights and biases. The selection of an optimization algorithm affects the convergence rate, stability of training, and the quality of the final solution.

The most basic approach is stochastic gradient descent (SGD), in which parameters are updated according to

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t), \tag{6.9}$$

where θ represents the set of all trainable parameters, η is the learning rate, and $\nabla_\theta \mathcal{L}$ is the gradient of the loss with respect to θ. While SGD is conceptually simple and computationally efficient, it can suffer from slow convergence and sensitivity to the choice of η.

Several extensions to SGD have been developed to improve performance. SGD with momentum accumulates a velocity vector in parameter space that smooths updates and accelerates convergence, particularly in regions with high curvature or noisy gradients.

Adam (adaptive moment estimation) combines the benefits of momentum with per-parameter adaptive learning rates. It maintains exponentially decaying averages of past gradients and squared gradients, allowing it to adjust the step size for each parameter individually. Its parameter update rule can be expressed as

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{6.10}$$

where $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of the first and second moments of the gradient, and ϵ is a small constant to prevent division by zero. Adam is widely adopted due to its robustness and minimal hyperparameter tuning.

RMSProp also adapts learning rates per parameter by maintaining a moving average of the squared gradients, effectively normalising updates and improving performance on non-stationary problems.

The choice of optimizer should be guided by the characteristics of the dataset, the scale of the network, and the sensitivity of the application to convergence stability. For many deep learning applications, Adam offers a good balance between convergence speed and stability, though SGD and RMSProp remain viable alternatives, particularly when fine control over generalisation behaviour is required. For this work, Adam is chosen.

6.1.5. Training

Once the concrete architectures of the autoencoder have been determined, the training takes place. As explained, the features that are taken from the LES results are time series, and are used to train the model. Initially, these features are all normalised between 0 and 1. This is done such that inherently high-valued features such as pressure and temperature do not dominate over smaller-valued features such as the numerous mass fractions. Furthermore, the features are split into training, validation, and testing groups. The training set is used to update the network weights, the validation set is used to monitor performance and prevent overfitting, and the testing set is reserved for the final evaluation of the trained model. The split used in this work is 80%,10%,10%, respectively, a common split.

During training, the network parameters are iteratively updated using the chosen optimizer to minimise the selected loss function, as described in the preceding sections. Each training iteration, or epoch, consists of a forward pass, where the input data are propagated through the encoder, bottleneck, and decoder, and a backward pass, where gradients are computed via backpropagation and used to adjust the weights and biases. However, a few parameters have to be determined manually to see which combination results in the lowest loss. These parameters are called hyperparamaters.

6.1.6. Hyperparameter Tuning

To optimise the architecture and training configuration of the autoencoder, a hyperparameter search was performed using the Optuna framework. The search objective was to minimise the validation loss, defined as the mean squared error between the input features and their reconstruction.

The hyperparameters explored covered both architectural and training choices. On the architectural side, the number and width of hidden layers in the encoder and decoder were selected from predefined, strictly decreasing sequences, with the constraint that the narrowest hidden layer remained wider than the latent dimension so that the intended bottleneck occurs at the latent code. The output activation was varied among linear, sigmoid, and ReLU: sigmoid is well matched to min-max normalized targets in $[0, 1]$, linear is appropriate for standardized or otherwise unbounded targets, whereas ReLU can introduce a zero floor and bias low-amplitude reconstructions. An optional L1 activity regularization term was applied to the latent layer to promote sparse, low-magnitude codes, which often yields more compact and separable latent clusters in the presence of multimodal dynamics (e.g., high- vs. low-frequency cycles). Finally, L1 and L2 kernel regularization weights on the layer matrices were tuned on logarithmic scales to control capacity and overfitting, with L1 encouraging weight sparsity (implicit pruning) and L2 providing smooth weight decay.

Training-related hyperparameters included the learning rate of the Adam optimiser (with AMSGrad enabled) and the batch size. The search process was guided by early stopping, learning rate reduction on plateau, and Optuna's pruning mechanism, which terminates underperforming trials based on in-

termediate validation loss. Each candidate model was trained for a maximum of 50 epochs, with the best-performing configuration selected based on the lowest observed validation loss. 50 epochs was chosen as at this point, the loss change was negligible.

This approach enabled a systematic exploration of the interaction between architectural depth, latent dimensionality, regularisation strength, activation functions, and optimisation settings, leading to a model configuration that balanced reconstruction fidelity with generalisation capability. Table 6.1 gives the summary of the hyperparameter turning variables.

Table 6.1: Summary of hyperparameters explored during Optuna search.

Hyperparameter	**Search Range / Options**	**Description**
Hidden layer widths	$\{12, 10, 8, 6, 4\}$ (strictly decreasing; depth ≤ 4)	Neurons per hidden layer in encoder/decoder; narrowest hidden layer must exceed the latent dimension.
L1 regularisation weight	$[10^{-8}, 10^{-5}]$ (log scale)	L1 penalty on layer weights to encourage sparsity.
L2 regularisation weight	$[10^{-6}, 10^{-4}]$ (log scale)	L2 penalty on layer weights (weight decay) to reduce overfitting.
Encoder activation	{ReLU, Sigmoid, Linear}	Activation for encoder hidden layers (mirrored in decoder).
Output activation	{ReLU, Sigmoid, Linear}	Activation applied to the decoder output layer.
Latent activity L1	$[10^{-8}, 10^{-4}]$ (log scale)	Optional L1 activity regularisation on latent outputs (promotes sparse codes).
Learning rate	$[10^{-4}, 10^{-3}]$ (log scale)	Adam optimiser step size.
Batch size	{16, 32, 64}	Samples per update.

Once the dimensionality reduction has been performed, the precursor identification process begins.

6.2. Precursor Identification

After the autoencoder has been created, the latent variables that are produced can be used for the precursor identification. This algorithm was developed by Golyska and Doan [46], and consists of mimicing a complex system as a weighted graph in the phase space. This approach can identify communities, which serve as precursor clusters belonging to an extreme event. This identification process is outlined in these sections.

6.2.1. Phase Space and Tesselation

To begin the identification, the latent variables are plotted in a multi-dimensional phase space diagram. As an example, a two dimensional phase space diagram is shown, with fictional data.

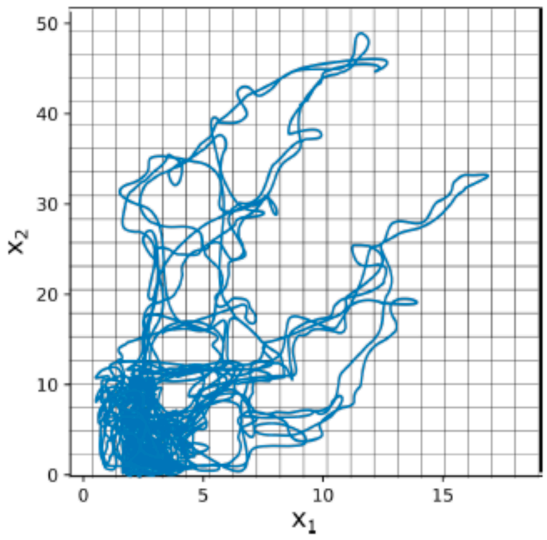


Figure 6.2: Phase Space Diagram with two variables [46]

In Figure 6.2, the evolution of two variables is depicted. This representation is inherently time-independent, serving solely to illustrate the progression of the variables within the phase space. When additional latent variables are considered, the dimensionality of the diagram correspondingly increases. To efficiently represent such higher-dimensional trajectories, tessellation is employed to compress the description of the system's evolution. To do so, the diagram is divided into M sections along each dimension, with the resulting regions referred to as hypercubes. The procedure begins by normalizing the phase space across all dimensions, thereby facilitating the identification of hypercubes containing points of the trajectory. Subsequently, the trajectory's time series in the normalized phase space is examined and converted into a sequence of hypercube indices. This transformation yields a discrete time series that maps the system's path by specifying the hypercube occupied at each time step. For computational efficiency, tessellation is implemented using sparse matrices, ensuring that only non-empty hypercubes are retained in memory, each assigned a unique index. This approach enables precise discretization of the system's trajectory, avoiding both overlaps and voids, as illustrated in Figure 6.3.

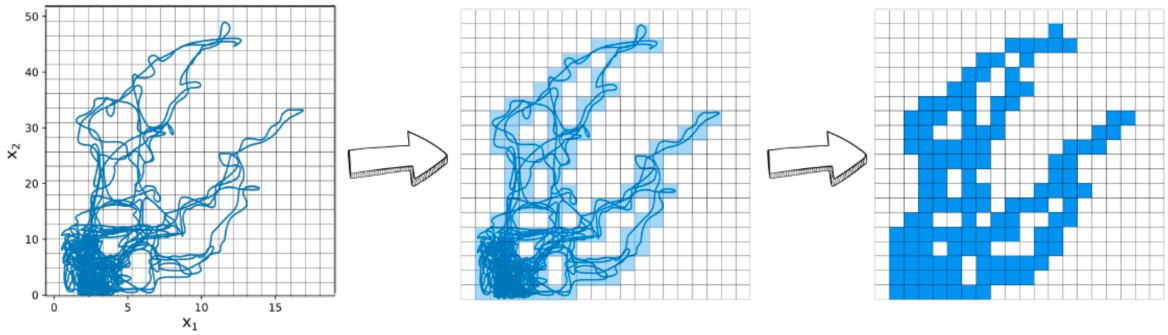


Figure 6.3: Phase Space Tessellation

Once the system has been tessellated, the transition probability matrix can created.

6.2.2. Transition Probability Matrix

Once the tessellation is complete, it must be known how the individual data points transition from one state to another. An example of such a matrix is shown in Figure 6.4

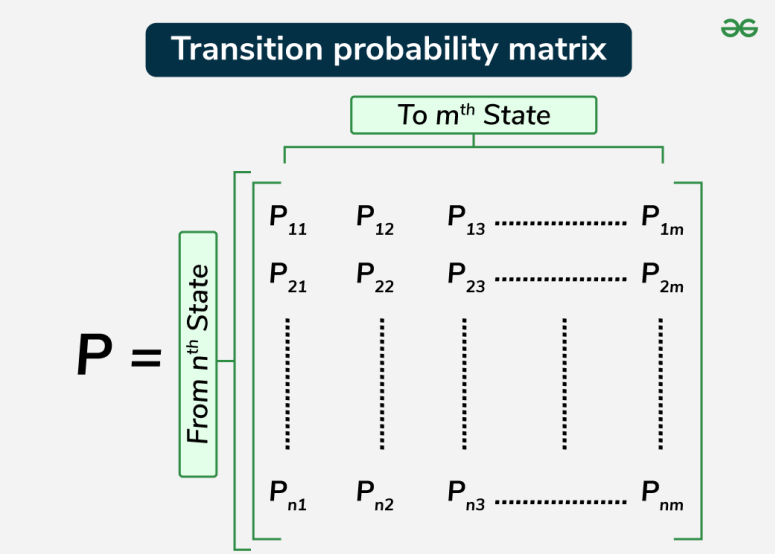


Figure 6.4: Transition Probability Matrix Format [45]

where in this case, the matrix indices indicate the probability of a hypercube in an n^{th} state transitioning to another in an m^{th} state. Normally, this is found using,

$$P_{ij} = \frac{m\left(B_i \cap \mathcal{F}^1(B_j)\right)}{m(B_i)}, \quad i,j = 1,\ldots,N \tag{6.11}$$

where $P_{i,j}$ represents the probability that the system transitions from hypercube B_i to hypercube B_j. Moreover, $m(B_i)$ indicates the number of phase space points in a hypercube B_i. The parameter N indicates the total number of hypercubes, while $\mathcal{F}^1$ denotes the temporal forward operator, which maps the current state of the system to its state at the next time step. However, the equation used in the work of Golyska and Doan [46] uses a backwards operator, shown as,

$$P_{ij} = \frac{m\left(B_i \cap \mathcal{F}^{-1}(B_j)\right)}{m(B_i)}, \quad i,j = 1,\ldots,N \tag{6.12}$$

Here, $\mathcal{F}^{-1}$ denotes the temporal backstep operator, which maps the current state of the system to its state at the preceding time step. The outcome of this step is a sparse transition probability matrix $\mathcal{P}$ of size M^n, where n is the number of dimensions of the phase space. A visualization of $\mathcal{P}$ is presented in Figure 6.5, which reveals that the matrix is predominantly diagonal, indicating that the trajectory remains within a given hypercube for several consecutive time steps. The non-zero off-diagonal entries correspond to transitions between different hypercubes.

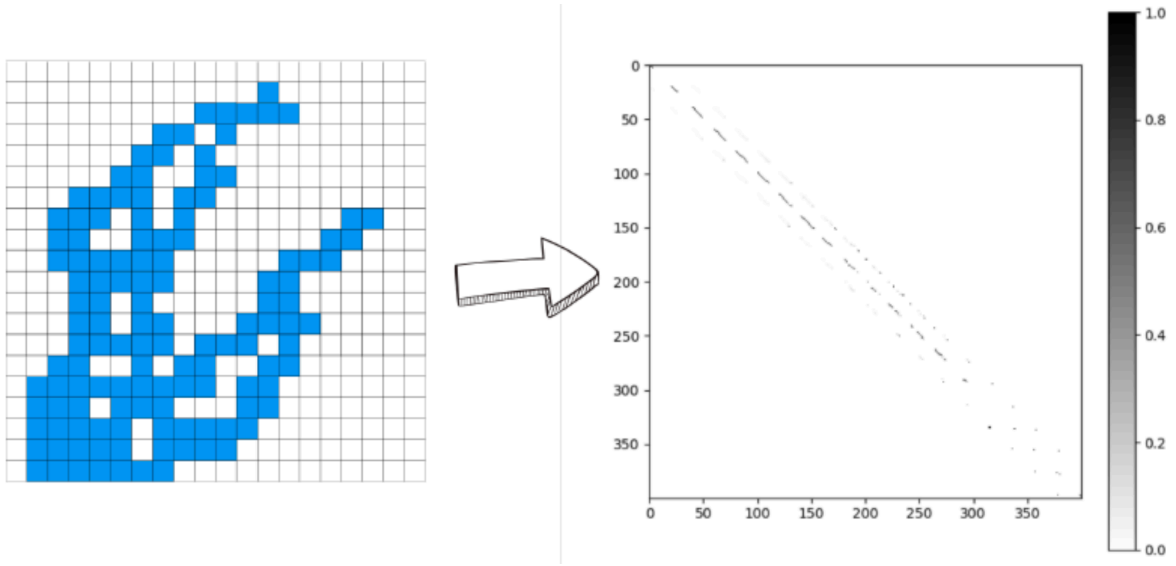


Figure 6.5: Tessellated Data to Transition Probability Matrix [46]

The transition probability matrix can be recast as a weighted, directed graph, where each node corresponds to a hypercube in the tessellated trajectory and each directed edge denotes a possible transition between hypercubes. The weight of an edge encodes the probability associated with the corresponding transition. An illustrative example of this is Figure 6.6,

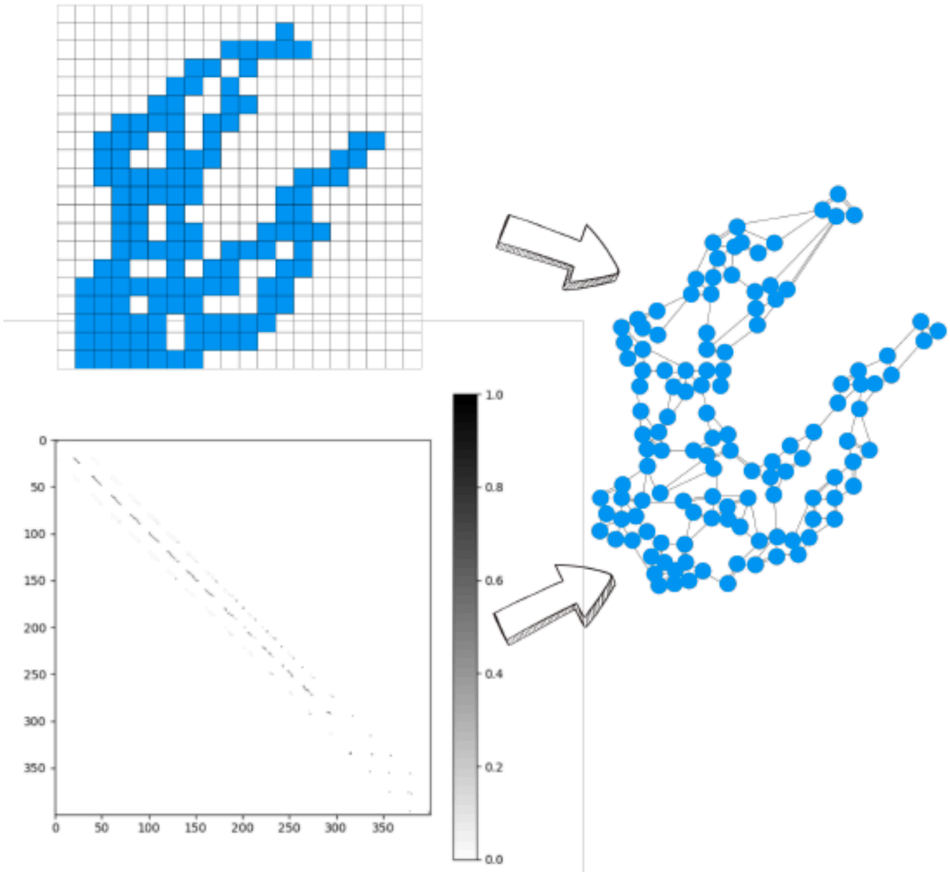


Figure 6.6: Weighted, Directed Graph [46]

6.2.3. Modularity

In this phase of the algorithm, the resulting network is partitioned into communities according to the modularity metric. Modularity quantifies the quality of a network division by measuring the difference between the proportion of intra-community edges observed and the proportion expected under a random null model. A high modularity value signifies that the inter-community edge count is substantially lower than expected by chance, thereby indicating a statistically significant community structure rather than a merely sparse interconnection.

The community detection procedure is implemented using the Python Modularity Maximization library [136], which follows the methodology proposed by Newman and Leicht [90, 77]. Furthermore, the computational efficiency enhancement introduced by Golyska and Doan [46] is retained in this implementation.

To create a modularity metric, the expected number of edges of a community is required. The expected number of edges is estimated by constructing a random network that preserves the degree sequence of the original graph. Each vertex i is assigned a degree k_i, interpreted as the number of incident half-links. The total number of half-links is therefore

$$\sum_i k_i = 2m, \tag{6.13}$$

where m denotes the total number of edges in the network.

A half-link from vertex i can connect to any of the remaining $2m-1$ half-links, excluding self-connections. For a vertex j with k_j half-links, the probability that one of i's half-links connects to j is given by

$$\frac{k_j}{2m-1} \approx \frac{k_j}{2m} \tag{6.14}$$

for large networks. Consequently, the probability that vertices i and j are connected is approximately

$$\frac{k_i k_j}{2m}. \tag{6.15}$$

For a given pair (i, j), the modularity, as defined in Equation 6.16, measures the deviation between the observed adjacency A_{ij} and its expected value $\frac{k_i k_j}{2m}$, where A_{ij} is the (i, j)-th entry of the adjacency matrix, equal to 1 if an edge exists and 0 otherwise. This formulation excludes multi-edges between the same vertex pair. Summing over all vertex pairs in a graph partitioned into two communities, with n denoting the number of vertices, yields the modularity expression:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{6.16}$$

where $\delta(c_i, c_j) = 1$ if vertices i and j belong to the same community, and 0 otherwise. In this formulation, s_i and s_j denote the community assignments of vertices i and j, respectively. The contribution to modularity arises exclusively from pairs of vertices within the same community, as indicated by the Kronecker delta function δ_{s_i,s_j}.

The modularity expression can be extended to account for the directionality of edges in a graph:

$$Q = \frac{1}{m} \sum_{ij} \left(A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m} \right) \delta_{s_i,s_j} \tag{6.17}$$

where k_i^{in} and k_j^{out} denote the in-degree and out-degree of vertices i and j, respectively. The probability of an edge from vertex j to vertex i in the corresponding model is $\frac{k_i^{\text{in}} k_j^{\text{out}}}{m}$. Unlike the undirected case in Equation 6.16, the factor of 2 does not appear in the denominator because the network is directed.

Following the method of Newman [90], the problem of modularity maximization can be reduced to partitioning the network into two communities. For computational convenience, Equation 6.17 is expressed in vector form as:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m} \right) (s_i s_j + 1) \tag{6.18}$$

which simplifies to:

$$Q = \frac{1}{2m} \sum_{ij} s_i B_{ij} s_j = \frac{1}{2m} \mathbf{s}^T \mathbf{B} \mathbf{s} \tag{6.19}$$

Here, $s_i \in \{+1, -1\}$ encodes the community membership of vertex i, and $\delta_{s_i, s_j} = \frac{1}{2}(s_i s_j + 1)$. The vector $\mathbf{s}$ collects all s_i values, while $\mathbf{B}$ is the modularity matrix with elements

$$B_{ij} = A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m}. \tag{6.20}$$

The optimization objective is to maximize Q for a given modularity matrix $\mathbf{B}$. In the case of directed graphs, $\mathbf{B}$ is generally asymmetric; therefore, symmetry is restored by adding its transpose. The resulting expression becomes:

$$Q = \frac{1}{4m} \mathbf{s}^T \left(\mathbf{B} + \mathbf{B}^T \right) \mathbf{s}. \tag{6.21}$$

This maximization problem is treated as an eigenvalue problem. The community assignment vector $\mathbf{s}$ is expressed as:

$$\mathbf{s} = \sum_l a_l \mathbf{v}_l$$

where $\mathbf{v}_l$ are the eigenvectors of $(\mathbf{B} + \mathbf{B}^T)$ and $a_l = \mathbf{v}_l^T \mathbf{s}$. Substituting into the modularity definition yields:

$$Q = \sum_l a_l \mathbf{v}_l^T (\mathbf{B} + \mathbf{B}^T) \sum_j a_j \mathbf{v}_j = \sum_l \beta_l \left(\mathbf{v}_l^T \mathbf{s} \right)^2 \tag{6.22}$$

where β_l and $\mathbf{v}_l$ are the eigenvalues and corresponding eigenvectors. The maximum value of Q is attained when $\mathbf{s}$ aligns with the eigenvector associated with the largest eigenvalue. Given the constraint $s_i = \pm 1$, the closest discrete approximation is selected. The signs of the components of the leading eigenvector determine community membership, enabling the bipartition.

The algorithm recursively divides the network into communities, stopping when no further increase in modularity is achieved. Instead of modularity itself, the change in modularity ΔQ is evaluated for each subdivision:

$$\Delta Q = \frac{1}{4m} \mathbf{s}^T \left(\mathbf{B}^{(g)} + \mathbf{B}^{(g)T} \right) \mathbf{s} \tag{6.23}$$

where

$$B_{ij}^{(g)} = B_{ij} - \delta_{ij} \sum_{k \in g} B_{ik} \tag{6.24}$$

and g denotes the subgraph under consideration. The process is iterative: after each division, the modularity matrix is updated (deflated), a new subgraph is formed, and the procedure repeats. Iterations continue until either the maximum iteration limit is reached or the number of communities falls below a user-specified threshold.

In the final steps of the clustering process, a community affiliation matrix is created. This ensures that the clusters identified in the previous step can be mapped to the vertices of the original graph. This matrix is then used to deflate the original probability transition matrix $\mathbf{P}$, producing a new transition matrix $\mathbf{P}^{(1)}$ that characterizes the dynamics of the updated network. This marks the point in the algorithm where the iterative procedure, as described in the preceding section, begins.

$$\mathbf{P}^{(1)} = \mathbf{D}^T \mathbf{P} \mathbf{D} \tag{6.25}$$

An illustration of both the original probability transition matrix and the deflated matrix is presented in Figure 6.7.

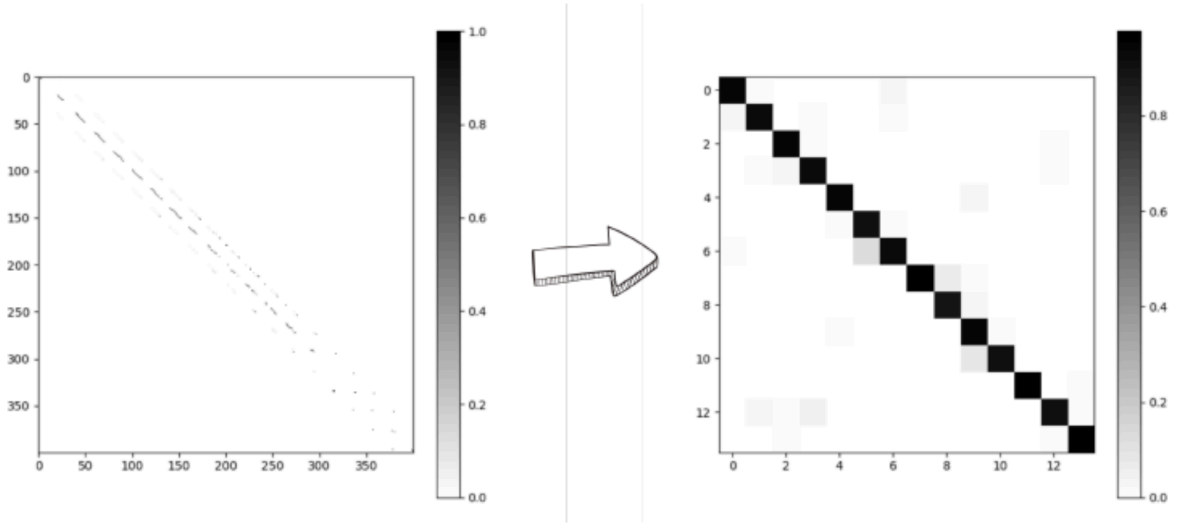


Figure 6.7: Deflation Matrix Visualization [46]

6.2.4. Cluster Classification

The final stage of the algorithm concerns the identification of extreme and precursor clusters. Extreme clusters are determined during the tessellation phase, in which hypercubes corresponding to extreme states are marked. Precursor clusters are then defined as those exhibiting transitions to extreme clusters, as indicated by the final transition probability matrix. Any cluster with a nonzero probability of transitioning into an extreme cluster is classified as a precursor cluster, irrespective of the magnitude of the probability. A visual representation of the identification procedure for both extreme and precursor clusters is provided in Figure 6.8 and Figure 6.9.

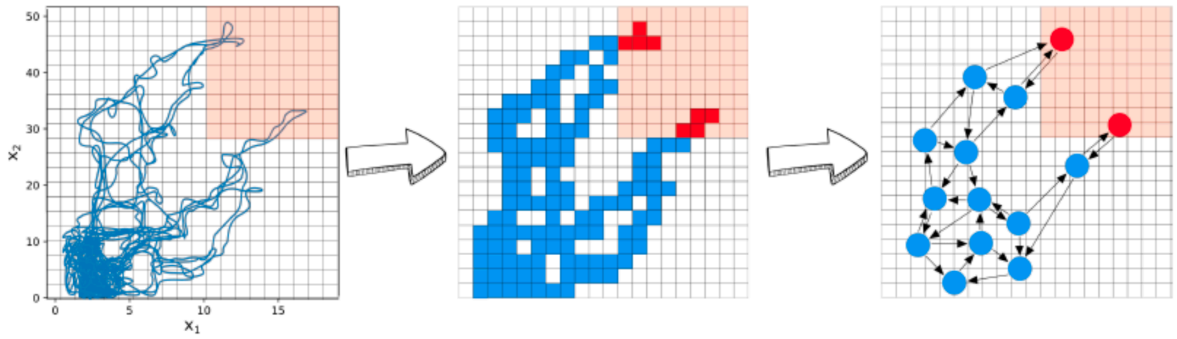


Figure 6.8: Extreme Event of Tessellated Data [46]

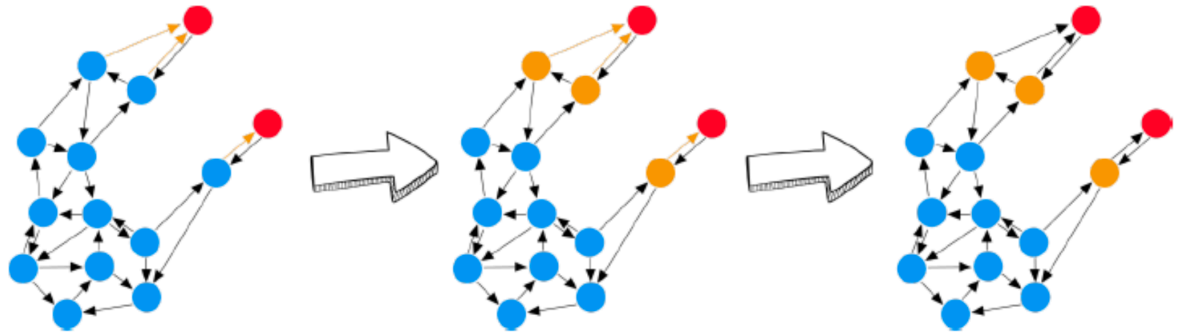


Figure 6.9: Precursor Clusters of the Weighted, Directed Graph [46]

Evidently, the definition of what constitutes an extreme cluster is of critical importance, as it determines when the time series transitions into an extreme state, thereby defining the onset of precursor clusters and the initiation of spray activation. In the work of Floris [42], a fixed threshold was employed to define extreme events, based on the temperature feature.

In the present study, however, normalized latent variables are utilized, which precludes the use of a physically grounded fixed threshold. Since flashback events are rapid and cyclic in nature, the latent variables exhibit sharp fluctuations analogous to those observed in the raw features. Consequently, the normalized value corresponding to the occurrence of these fluctuations is adopted as the defining threshold for extreme clusters. This threshold is further validated through the robustness analysis to ensure its reliability.

6.3. Robustness Testing

This section outlines the tests conducted to evaluate whether the modularity-based clustering analysis can be applied reliably across different scenarios.

First, the sampling locations are systematically varied. As previously discussed in subsection 5.2.2, this is done to assess the effect of sampling depth within the mixing duct. Each location yields a distinct set of results, and by varying these locations, it becomes possible to extract more informative and representative features from the dataset.

Although one location is selected as the primary sampling site, a second location is reserved as a robustness benchmark. Features from this robustness site are passed through the trained autoencoder to assess out–of–sample reconstruction and generalization to previously unseen data. The resulting latent representations are then subjected to the same clustering procedure, providing an independent check on site selection: if reconstruction quality and clustering performance at the robustness site are comparable to, or exceed those at the primary site, the alternative location may be preferable; if they degrade materially, the original choice is validated.

Next, the input to the autoencoder consists of a diverse set of 14 features, each capturing different aspects of the system's behaviour. While the architecture of the network is optimised, the number of latent dimensions is treated as a free parameter and is varied between 2 and 4. This choice allows for an assessment of how the dimensionality of the latent space influences the performance of the clustering algorithm. Although using four latent dimensions may encode slightly richer information than three, the improvement in clustering quality may be marginal while incurring a significantly higher computational cost. Also for comparison, and following the methodology of Floris [42], the selected features are also provided directly to the clustering algorithm in their raw form. This parallel approach

enables a direct evaluation of clustering accuracy between the autoencoder-based representation and the baseline used in prior work.

Also, the definition of the extreme event threshold is varied. As established, the identification of extreme clusters plays a central role, since it determines when precursor clusters are formed and when the mitigation strategy (sprays) is activated. In Floris [42], the temperature feature allowed for the use of a fixed, physically motivated threshold. In the present work, however, normalized latent variables are employed, which precludes the direct use of a physics-based cut-off value. Instead, the threshold is defined based on the normalized latent values corresponding to the rapid fluctuations observed during flashback events. By changing this defining value and verifying it through robustness analysis, the sensitivity of the clustering performance to threshold selection can be assessed.

Finally, the robustness test conducted for the modularity-based clustering algorithm evaluates its predictive capability in an online setting with limited data availability. Owing to the substantial computational cost associated with converging the probability transition matrix, the algorithm is unable to perform clustering in real-time, i.e., it cannot recompute the probability transition matrix upon the arrival of each new data point in the time series. Consequently, it becomes important to assess whether the algorithm possesses sufficient robustness to reliably predict the behaviour of previously unseen data, based solely on patterns learned from past time series.

To investigate this, the dataset containing the time series of the selected features is partitioned into training and test subsets. The clustering procedure is first applied to the training set, producing clusters corresponding to normal, extreme, and precursor states, along with the associated probability transition matrix. The centroids of these clusters are then recorded and subsequently employed for classification of the test set. Specifically, the state of each new data point in the test sequence is determined by computing its distance to the pre-identified cluster centroids, according to the following formula:

$$\epsilon = \sqrt{\sum_{i=1}^{n} \left(\tilde{\phi}_i - \tilde{\phi}_{i,\text{cluster}} \right)^2} \tag{6.26}$$

where $\tilde{\phi}_i$ denotes the current state of the incoming data, n is the number of features, and $\tilde{\phi}_{i,\text{cluster}}$ represents the centroid of the corresponding cluster obtained from the training set. To properly account for the influence of the different features, both the test data states and the cluster centroids are normalized using min–max normalization.

Once the closest cluster has been identified, the current state is assigned the label of that cluster. A precursor to an extreme event is thus detected whenever the data point is classified into a precursor cluster. In this manner, the approach enables an online prediction of system behaviour and provides an assessment of the robustness of the clustering algorithm. Furthermore, this test also serves as a useful indicator of the method's performance when only shorter time series are available.

7

Outcome

In this chapter, results addressing the research questions are presented and discussed. First, the LES outcomes and observed phenomena are summarized with emphasis on autoignition and flashback behaviour. Next, dimensionality-reduction results are analysed to justify the final feature and latent-dimension choices. The clustering-based segmentation and precursor detection are then evaluated, including robustness tests on unseen locations and alternative latent dimensions. Finally, the features of Floris [42] are given to the clustering algorithm as a test, and the water spray results are presented as a form of flashback suppression.

7.1. LES

In this section, the results of the 3D LES of the Ansaldo Energia GT36 are presented. The methodology employed is described in Section 5.1. Before analyzing the combustion dynamics, the quality of the LES must be assessed to ensure it has run as expected. For this purpose, Pope's criterion is applied, as introduced in Equation 5.41. While Tecplot provides the subgrid-scale kinetic energy directly as a standard variable, K, the turbulent kinetic energy of the resolved scales must be computed as

$$K = \tfrac{1}{2}\left(\tilde{u}^2 + \tilde{v}^2 + \tilde{w}^2\right), \tag{7.1}$$

where $\tilde{u}$, $\tilde{v}$, and $\tilde{w}$ are the time-averaged RMS values of the perturbations of the velocity components. Using this variable, Pope's criterion can be evaluated across the entire mesh, where values of 0.2 or lower are considered acceptable for a high-quality LES. The results at a representative timestep are shown in Figure 7.1.

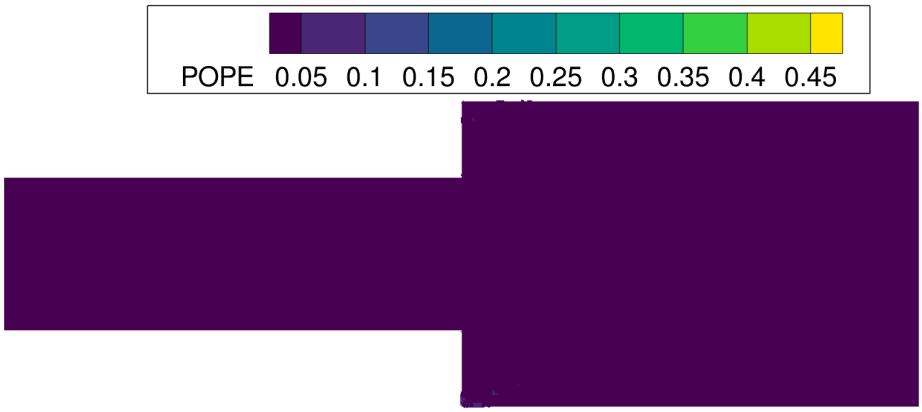


Figure 7.1: Pope's Criterion for the 3D LES

As can be seen, nearly all regions of the LES exhibit values approaching zero, which demonstrates that most of the turbulent kinetic energy is resolved rather than modeled. This confirms that the LES quality is excellent. However, certain localized regions within the outer recirculation zone (ORZ) display slightly higher values, with some small zones surpassing 0.2, as is evident from the legend. These elevated values can be attributed to the strong velocity gradients and wall-bounded shear layers present in this region, which naturally enhance the subgrid-scale contribution. Since these zones are spatially confined and intermittent, the overall LES quality remains robust.

7.1.1. Flashback Evolution

The flame structure obtained in this LES can be compared with the works of Floris [42], Rouco Pousada et al. [104], and Kruljevic et al. [73]. These studies identify two key flow regions: the central developing zone (CDZ), a high-momentum core jet that issues from the swirler and convects downstream before interacting with recirculation zones, and the outer recirculation zones (ORZ), which form near the combustor walls due to flow separation and swirl-induced corner vortices. Both CDZ and ORZ exhibit similar spatial extents and dynamics to those observed here, including a shear layer separating them, as shown in Figure 7.2. This agreement provides additional confidence in the predictive capabilities of the present LES, as the large-scale flow organization and flame anchoring mechanisms are consistent with prior experimental and numerical investigations.

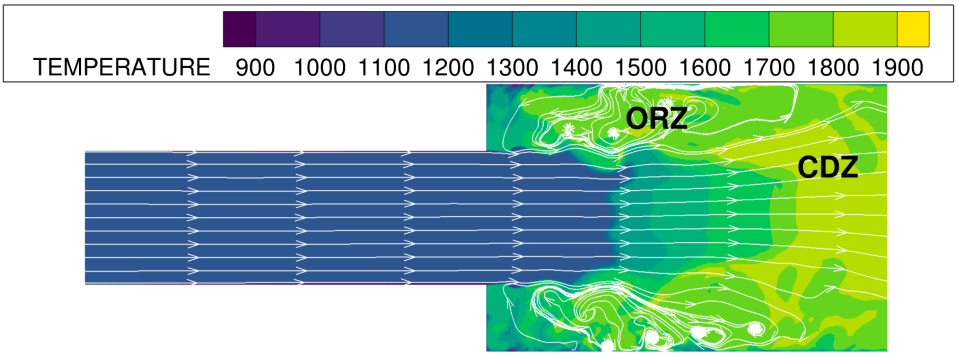


Figure 7.2: Flame Shape

The flashback process is illustrated in Figure 7.4. Combustion begins at $t = 1\,\mathrm{ms}$, after the inert flow has fully developed. Autoignition first occurs near the expansion step, where the balance between ignition delay time and flow-through time anchors the flame base [panel (a)]. The initial heat release generates

a strong pressure wave (Figure 7.3), which propagates both downstream and upstream, reflects off the chamber walls, and interferes constructively along the centerline. The resulting compressive heating raises the local temperature, producing positive fluctuations that accelerate a secondary autoignition event [panel (b)].

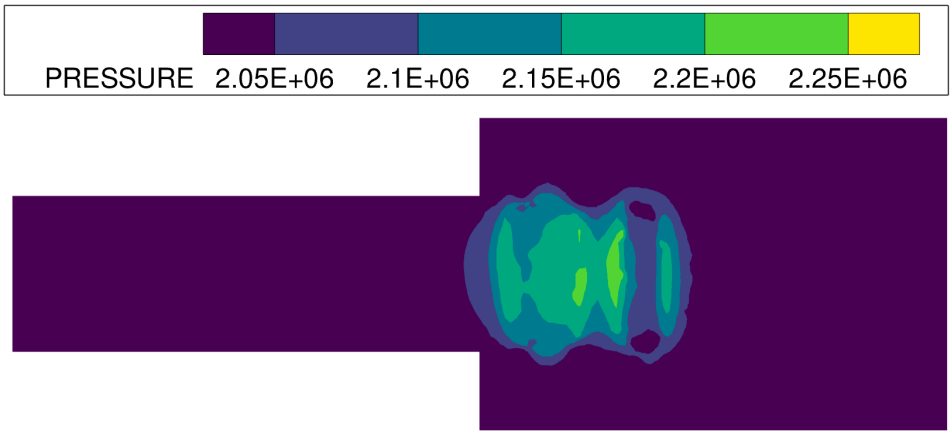


Figure 7.3: Pressure Rise at Ignition Kernel

The secondary autoignition launches another upstream-traveling pressure wave, imposing an unfavorable gradient on the incoming flow. This so-called “piston effect” compresses and heats the reactants, shortening the autoignition delay throughout the mixing duct and enabling kernels to form further upstream [panel (c)]. The boundary layer is especially susceptible due to its low velocities and longer residence times, which allow kernels to survive and propagate, ultimately causing boundary-layer flashback. The flame then advances upstream, nearly reaching the inlet, where the shortened ignition delay matches the local residence time under elevated pressures and temperatures [panel (d)].

Once the pressure wave reaches the inlet, compressive forcing ceases and the system enters a relaxation phase. Pressure and temperature return toward baseline values, and the flame is convected downstream, often stabilizing at a position further downstream than its original anchoring point. The cyclic nature of this process is evident: ignition kernels reappear in the recirculation zones, repeating the sequence of autoignition, pressure-wave generation, compressive heating, boundary-layer flashback, and relaxation.

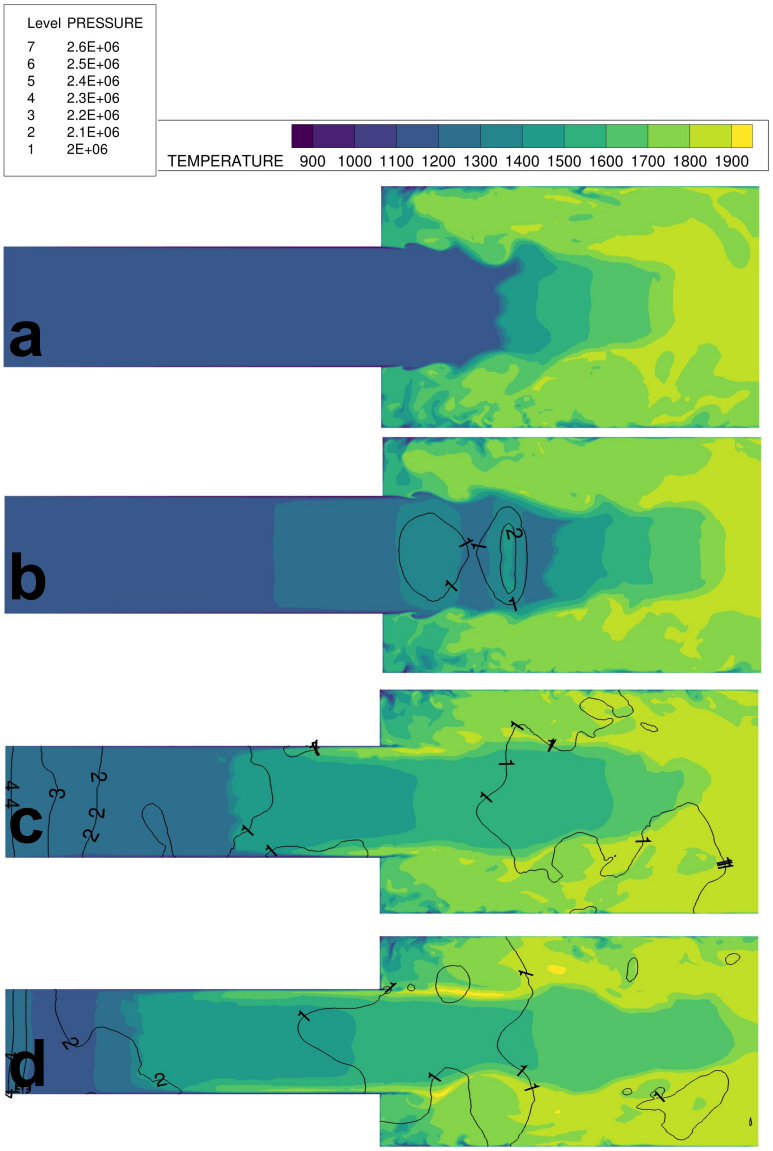


Figure 7.4: Flashback Evolution

Autoignition, the dominant propagation regime for the GT36, is highly sensitive to temperature. As discussed in subsection 3.1.4, the autoignition delay strongly depends on temperature near the cross-over point. With an inlet temperature of 1180 K, only a small increase is sufficient to shift the flame position. The cross-over temperature of roughly 1350 K is quickly reached once ignition kernels form, as observed slightly upstream of the step before full autoignition occurs in the premixing tube.

The pressure wave in this configuration attains amplitudes up to 26 atm at the inlet, propagating at nearly 650 m/s (the local speed of sound in the mixture). This results in a pronounced piston effect, reducing the incoming flow velocity from about 200 m/s to below 100 m/s, as shown in Figure 7.5. The effect intensifies with larger equivalence ratios and higher inlet velocities. Contrary to expectations, higher bulk velocities here amplify compressive heating, thereby facilitating autoignition. Equivalence ratios above $\phi = 0.2$ have been shown to generate stronger pressure waves [50].

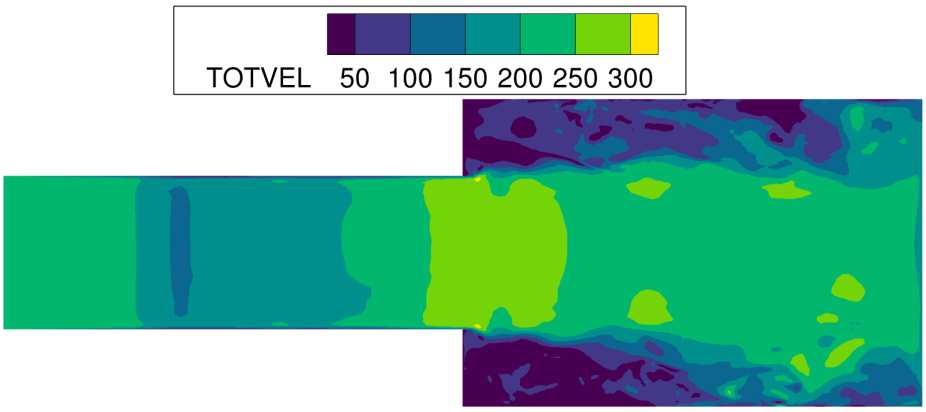


Figure 7.5: Piston Effect on Velocity Magnitude

Finally, boundary layer flashback further promotes upstream propagation through localized wall heating. It occurs when the turbulent burning velocity exceeds the local near-wall flow velocity, aided by vortical structures that induce boundary layer separation. For hydrogen, the risk is especially high due to its very small quenching distance of 0.64 mm. Under the present conditions, the laminar flame speed is about 13 m/s, but turbulence enhances the effective turbulent flame speed to above 60 m/s for an inlet turbulence intensity of 0.1 [50]. This, combined with hydrogen's reactivity, strongly favors wall-adjacent propagation. BL flashback differs from autoignition-driven movement, as it arises from flame front propagation rather than compression heating. In Figure 7.4(d), near-wall flame propagation surrounded by autoignition fronts confirms the coexistence of both mechanisms. In this LES, the piston effect and reduced velocities near the wall create favorable conditions for BL flashback, which in turn elevates local temperature and pressure, accelerating subsequent autoignition and aiding upstream flame movement.

7.1.2. Mass Fractions

To further understand the dynamics of the flashback, certain mass fractions are observed. These were chosen specifically due to their attributes that can provide insight about the flashback regime. As the sampling point can be 1 of 4 in Figure 5.8), the far-right point S3 will be taken as a measurement here due to its vicinity to the flame front, but slightly further distance compared to S4 to allow for a longer relaxed state.

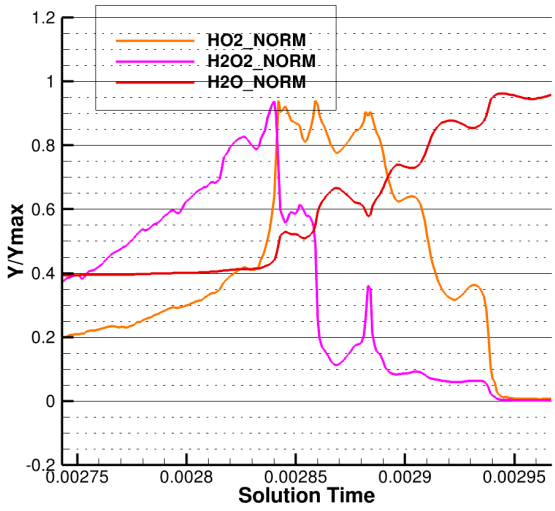


Figure 7.6: Normalized Mass Fraction of Species H_2O,H_2O_2,HO_2

The temporal evolution of the normalized mass fractions of HO_2, H_2O_2, and H_2O was extracted at a probe located just inside the boundary layer along the upper wall, immediately downstream of the sudden expansion. The near-wall region is characterized by reduced axial velocity, longer residence time, enhanced heat loss to the wall, and strong shear; all four effects modulate both the pre-ignition chemistry and the transition to stable combustion. Prior to $t \approx 0.00284\,\mathrm{s}$, the mixture at the point remains in a low-temperature regime. In this stage, HO_2 and H_2O_2 accumulate steadily while H_2O remains nearly constant, reflecting inhibited chain-branching below the crossover temperature. When the local temperature crosses the crossover threshold at $t \approx 0.00284\ \mathrm{s}$, raised in this case by compressive heating associated with the previously discussed pressure wave, the chemistry undergoes a qualitative transition. The accumulated H_2O_2 rapidly decomposes,

$$\mathrm{H_2O_2} \longrightarrow 2\,\mathrm{OH} \tag{7.2}$$

and the resulting OH radicals accelerate key branching and consumption pathways, exemplified by

$$\mathrm{H_2 + OH} \longrightarrow \mathrm{H_2O + H} \tag{7.3}$$

$$\mathrm{H + O_2} \longrightarrow \mathrm{OH + O} \tag{7.4}$$

which collectively precipitate the observed surge in H_2O. At the same instant, H_2O_2 collapses and HO_2 exhibits a sharp peak, marking the ignition transition at the wall-adjacent location.

Unlike centerline probes, the near-wall signal does not relax monotonically after ignition. Instead, HO_2 shows pronounced post-ignition oscillations and intermittent persistence. These features arise from boundary-layer transport: the low axial velocity increases the local residence time, permitting pockets of relatively colder mixture to survive within the shear layer, while wall heat losses and small-scale vortices intermittently quench and re-ignite the radical pool. Hydrogen's small quenching distance further accentuates this behaviour, so that HO_2 can be periodically replenished even as H_2O_2 remains depleted. Meanwhile, the steady rise of H_2O indicates the establishment and subsequent advance of a deflagration layer along the wall, consistent with boundary-layer flashback. Therefore, the observation at this specific point captures the coupled mechanism by which crossover-triggered autoignition, amplified by compressive heating, seeds near-wall flame propagation; the boundary layer then sustains and modulates the process through extended residence time, shear-induced mixing, and wall-mediated

heat transfer.

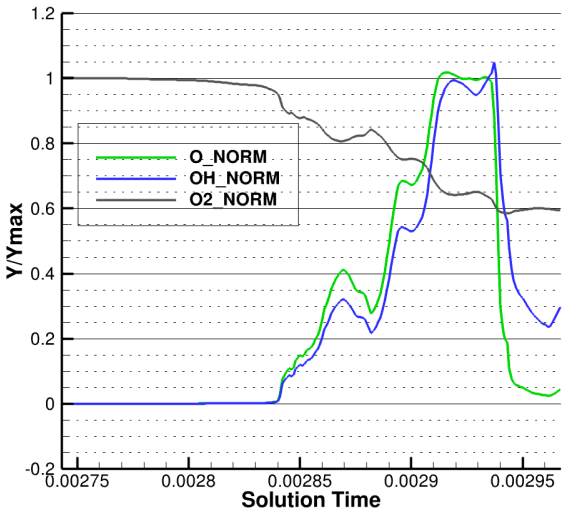


Figure 7.7: Normalized Mass Fraction of Species O,O_2,OH

In Figure 7.7, prior to $t \approx 0.00284$ s, both O and OH remain negligible, while O_2 persists near its maximum normalized value. This regime corresponds to the pre-ignition stage, dominated by low-temperature chemistry in which HO_2 and H_2O_2 act as the primary radical carriers. Radical generation is strongly suppressed, consistent with the extended induction period near the wall where residence times are longer and heat losses are significant.

Once the crossover temperature is reached, however, a sharp rise in O and OH is observed around $t \approx 0.00285$–0.00290 s. This marks the transition from low-temperature to high-temperature chemistry, driven primarily by the rapid decomposition of hydrogen peroxide as in Equation 7.2, which releases highly reactive OH radicals. The simultaneous growth of O is linked to classical chain-branching reactions such as Equation 7.4, and the subsequent radical recycling through

$$\mathrm{O} + \mathrm{H_2} \longrightarrow \mathrm{OH} + \mathrm{H}.$$

Together, these reactions create a strongly coupled O/OH radical pool that accelerates ignition and sustains high reactivity in the system.

During this stage, O_2 begins to decrease steadily, reflecting its consumption through radical-driven oxidation pathways. The rise and subsequent oscillations of O and OH can be attributed to the near-wall location of the probe. Here, the extended residence time within the boundary layer promotes localized radical accumulation, while wall heat transfer intermittently quenches radical growth. This competition manifests as fluctuations in the radical pool, indicating that ignition near the wall is subject to both autoignition chemistry and deflagration-like propagation effects.

7.1.3. Sampling Points

It was previously noted that wall-adjacent sampling locations may behave differently. Therefore, a robustness analysis is required to identify the most suitable location for data extraction (see Figure 5.8). The data should meet the following criteria:

1. **Cycles:** The time series should capture the flashback dynamics, exhibiting clear cyclic behavior with pronounced excursions (positive or negative) from the nominal level.

2. **Noise:** Because the wall region is more turbulent than the centerpoint, some noise is expected. While informative, this noise should not be so large that it impairs the model's ability to form coherent clusters.
3. **Relaxation:** Sampling locations at different streamwise positions experience different flashback durations. Very long flashback periods with minimal relaxation can blur the distinction between normal and precursor clusters, whereas very short flashbacks with long relaxation may provide too little information about flashback behavior.
4. **Regularity:** More regular temporal patterns, both in cycle shape and in the magnitudes reached and then recovered, improve learnability and typically yield better predictive performance.

To perform an analysis, a thermodynamic property, a velocity vector, and a mass fraction will be evaluated to be diverse. Initially, the temperature was observed. These are shown in Figure 7.8

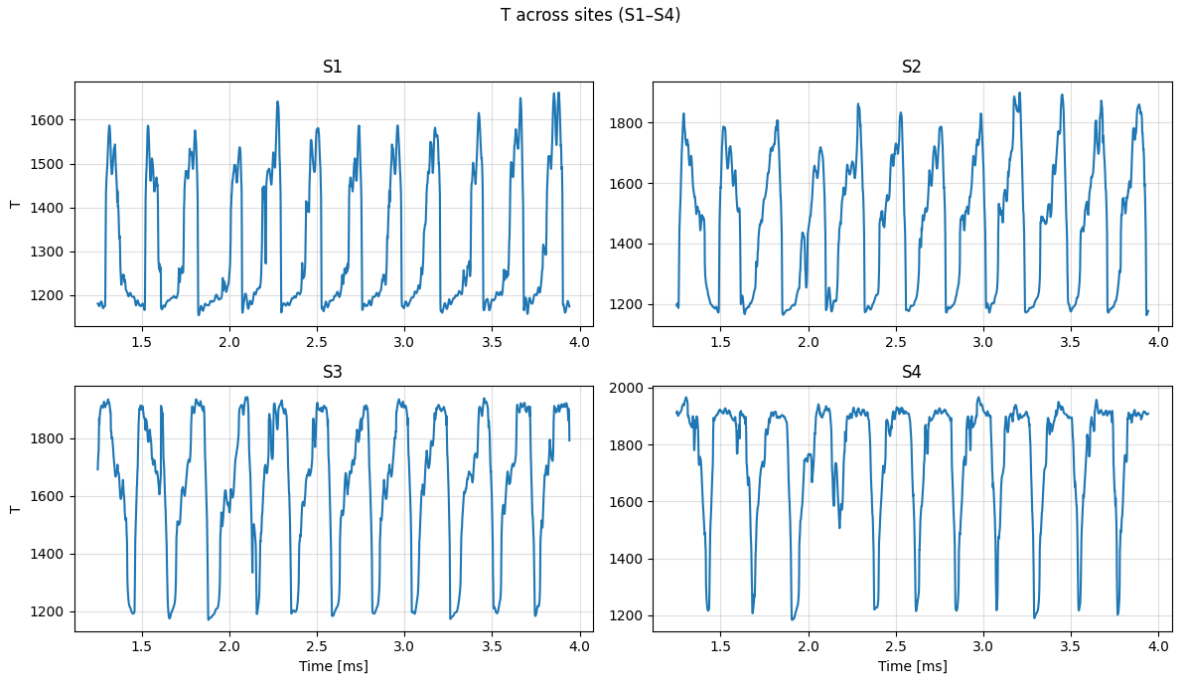


Figure 7.8: Temperature Extraction at Different Points

The temperature histories at the four probes exhibit clear flashback cycles at all locations. Cycle prominence increases toward the flame: S3–S4 routinely attain hot-plateau levels of $\sim$1900–1950 K, whereas S1 often peaks only at $\sim$1550–1600 K and at times does not fully enter the hot state. Noise is present everywhere but is proportionally largest in S1, whose cool-phase baseline shows higher-frequency variability; by contrast, S3–S4 display comparatively smooth hot plateaus, which should facilitate cluster separability. Relaxation behavior varies systematically with position: S1 exhibits long cool intervals punctuated by short hot excursions, while S4 remains hot for longer with brief relaxation phases (S2–S3 are intermediate). In terms of regularity, S3–S4 show the most consistent periods and peak magnitudes, whereas S1 exhibits stepwise ramps and occasional irregular cycles. However, the relaxation state is better for the further locations, and quite short for S4, which almost shows instant jumps after relaxation. Furthermore, the velocity u_x can be observed.

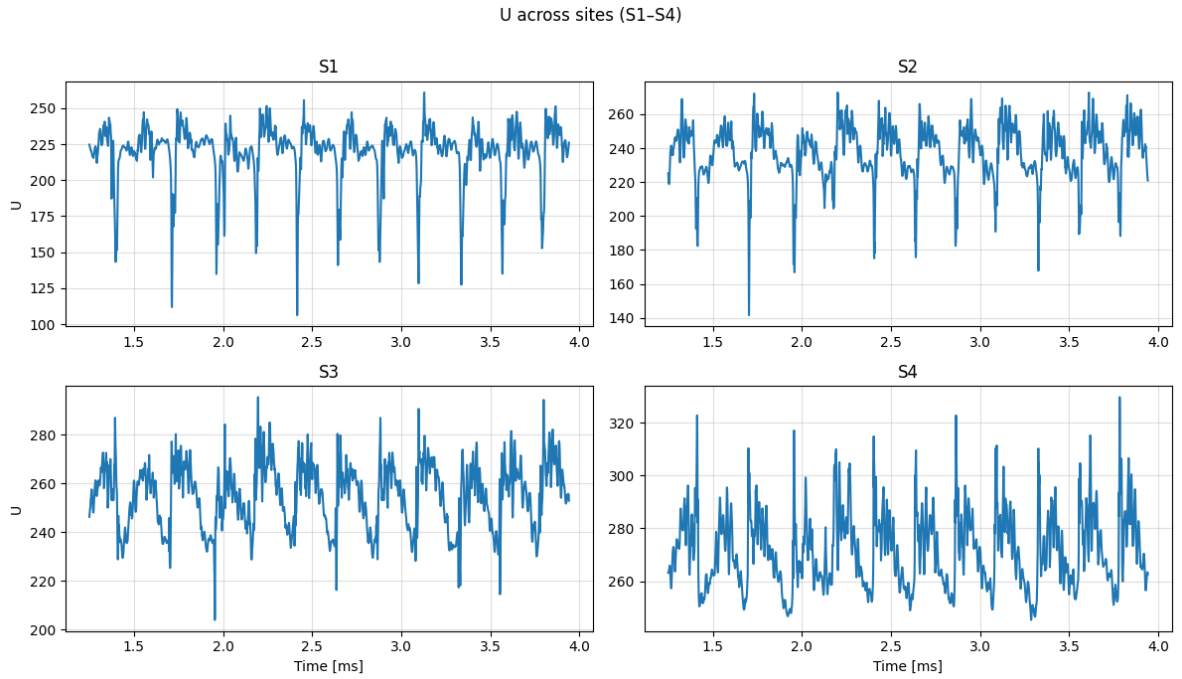


Figure 7.9: Velocity Extraction at Different Points

The cyclicity of u_x is expressed as periodic decelerations from a quasi-steady baseline. Cycle amplitude and clarity increase toward the flame: S3–S4 display deeper and more repeatable troughs with rapid recoveries, whereas S1 shows shallower excursions and occasional cycles that do not fully decelerate. High-frequency noise is present at all locations but is proportionally largest at S1; generally, all locations present clean extrema and smooth inter-event segments, which should aid cluster separability. As for relaxation, similarly, the difference in time spent in the high-u_x (relaxed) state is negligible between the locations. Regularity is highest at S3–S4, which exhibit more consistent inter-event spacing and trough depths. Finally, a mass fraction can be evaluated.

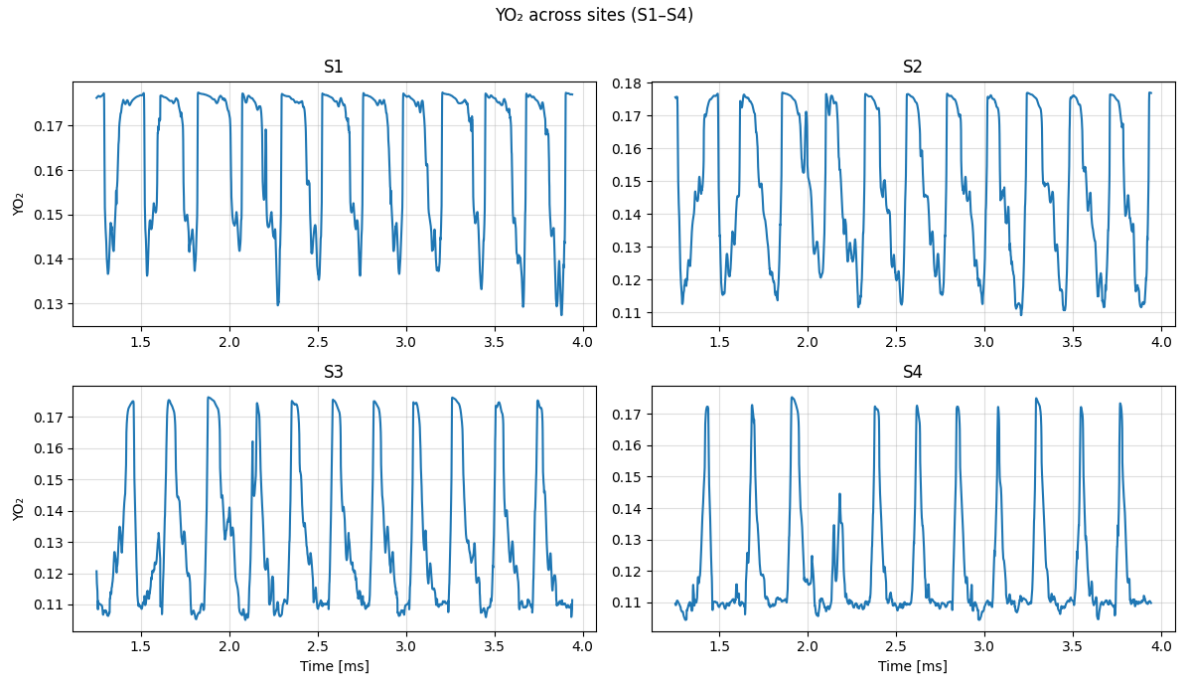


Figure 7.10: Oxygen Mass Fraction Extraction at Different Points

The O_2 mass–fraction has cycles whose clarity increases toward the flame. At S1, the series stays near a high-O_2 baseline ($\sim$0.17–0.18) with intermittent, relatively shallow depletions ($\sim$0.13–0.15), yielding weaker contrast and a comparatively ragged baseline. S2 shows deeper, more frequent depletions (down to $\sim$ 0.11–0.12) with clearer recoveries. Closer to the flame, S3 and S4 display the strongest cyclic signature: extended low-O_2 phases ($\sim$ 0.1–0.11) followed by brief returns to the high state, with sharp extrema and consistent period and amplitude. Noise levels are modest overall but smallest at S3–S4, whose repeatable trough depths and inter-event spacing should enhance cluster separability. Relaxation duration varies monotonically with position: S1 exhibits long relaxed (high-O_2) intervals and short depletions, whereas S4 shows the opposite, with S2–S3 intermediate.

Considering all the criteria (cycles, noise, relaxation, regularity), for each type of feature, S3 and S4 would provide the most insight about the flashbacks. However, the relaxation state of S4 is very short and may not give enough information to the clustering algorithm about a normal state; therefore S3 will be used.

7.1.4. Features

Based on the results provided in this section, and the analysis performed on the LES, the following 14 features will be used moving forward:

1. P (Pressure)
2. ρ (Density)
3. T (Temperature)
4. U (Velocity in the x-direction)
5. V (Velocity in the y-direction)
6. W (Velocity in the z-direction)
7. Y_H (Mass fraction of H radical)

8. Y_{H_2} (Mass fraction of molecular hydrogen)
9. Y_{H_2O} (Mass fraction of water)
10. $Y_{H_2O_2}$ (Mass fraction of hydrogen peroxide)
11. Y_{HO_2} (Mass fraction of hydroperoxyl radical)
12. Y_O (Mass fraction of atomic oxygen)
13. Y_{O_2} (Mass fraction of molecular oxygen)
14. Y_{OH} (Mass fraction of hydroxyl radical)

7.2. Dimensionality Reduction

Furthermore, to condense these 14 variables into workable data for the clustering algorithm, a dimensionality reduction will be used. As the latent variables are ranged from 2-4, 3 different optimisations have been run to obtain the optimal solution for each situation. A few parameters are concrete and do not change for the simulations, shown in Table 7.1. Once the trainings have been performed, the

Table 7.1: Training configuration and autoencoder architecture

Parameter	Value
Epochs	50
Loss function	MSE
Input/Output dimensions	14 (first encoder and last decoder layers)

autoencoder's decoder output is a full reconstruction of the original input time series, which can be qualitatively compared to the inputs, as an understanding of how the latent variables are constructed. The thermodynamic and velocity based features are shown in Figure 7.11.

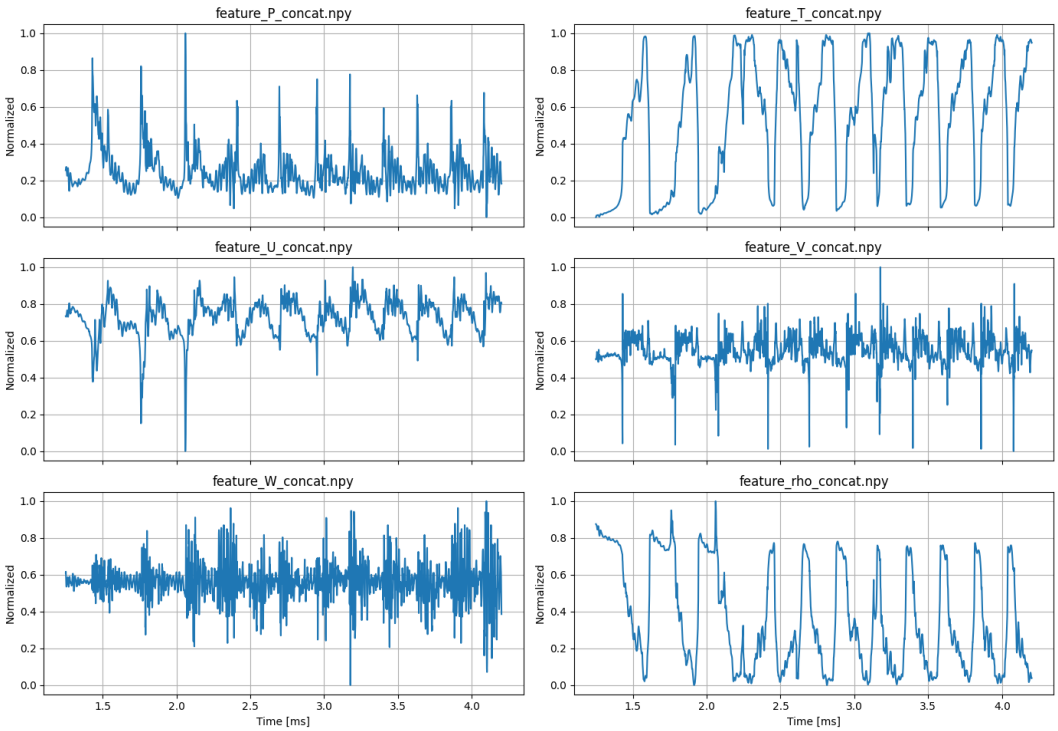


Figure 7.11: LES Extracted Thermodynamic and Velocity based Features

Across the features, common trends coexist with clear distinctions. Pressure and the transverse velocity components (v and w) are dominated by high-frequency noise with large excursions at flashback instants; this variability is unlikely to aid clustering and may hinder identification of precursor clusters. By contrast, density and temperature exhibit nearly identical cyclic structure, apart from the expected sign inversion (temperature rises coincide with density drops), which is suited for precursor detection. The streamwise velocity u combines both behaviors, showing discernible cycles superposed with noise. Because all signals were sampled at S3, their relaxation durations and cycle regularity are aligned, reducing variation and therefore benefiting the algorithm. The mass fractions were also extracted and are shown in Figure 7.12.

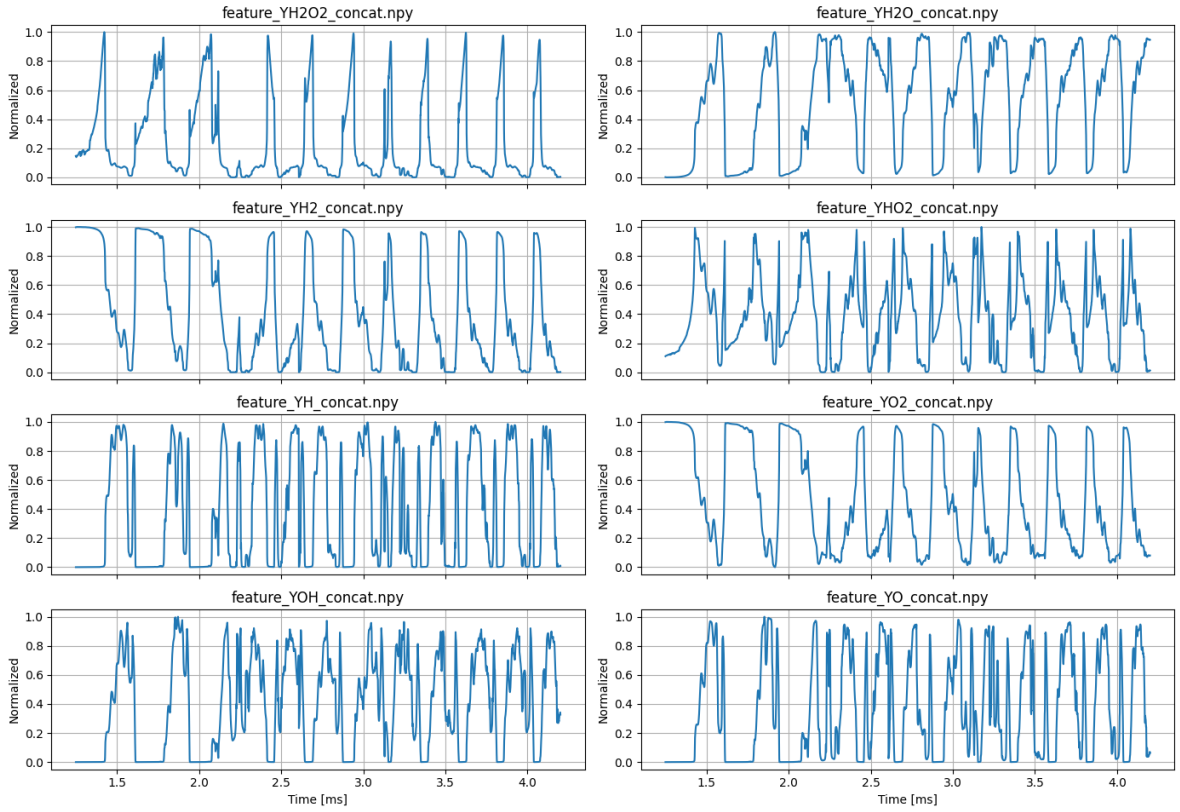


Figure 7.12: Mass Fraction Features

Relative to Figure 7.11, the mass-fraction features are more mutually similar, exhibiting pronounced cyclic patterns whose frequency varies modestly across species. Certain species (e.g., OH) display greater high-frequency fluctuations than others (e.g., H_2O); nevertheless, both retain a clear flashback signature that is informative for analysis. Many species alternate between active and relaxed states at either high or low concentration levels; this bimodality is consistent across cycles and, rather than posing difficulties, is unlikely to impede (and may even facilitate) precursor identification.

As a summary of the trainings, their train/validation/test losses were compared in Figure 7.13. The trend is monotonic: increasing the latent dimension lowers loss on *all* splits. Moving from 2→3 latents yields the largest drop (train: $3.7\times10^{-3} \rightarrow 2.9\times10^{-3}$, val/test: $3.1\times10^{-3} \rightarrow 2.3\times10^{-3}$), indicating that a 2D code is under–parameterised for the data's variability. The step from 3→4 latents brings a smaller, but consistent, refinement (val/test to $\approx 2.0\times10^{-3}$), suggesting diminishing returns beyond three degrees of freedom. Across all settings the ordering train $>$ val $\approx$ test persists, with validation never exceeding

training, consistent with a chronological split in which later segments are slightly “cleaner,” and with no evidence of overfitting. In practical terms, three latents capture most of the recoverable structure; a fourth latent mainly polishes mid–frequency details, while not producing further gains.

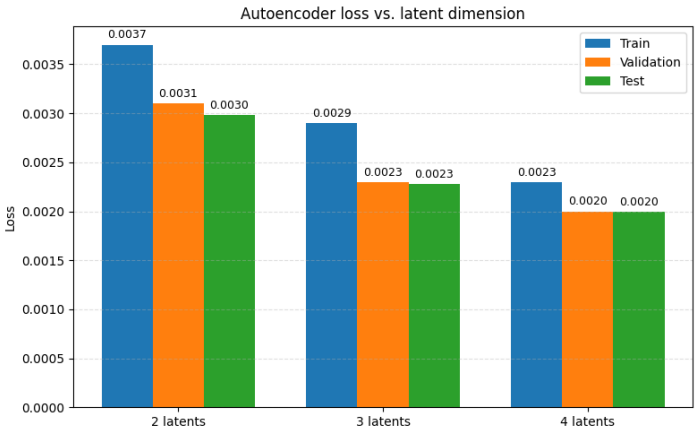


Figure 7.13: Comparison of Different Losses of Autoencoder Structures

After this figure, the details of the best configuration, the 3 latent structure, is provided in the subsections relating to the optimisations. The discussion of the 2 and 4 latent variable structures can be found in Appendix D and Appendix E respectively.

7.2.1. 3 Latent Variables

The optimal optimisation results are shown with this configuration.

Table 7.2: Summary of hyperparameters explored for 3 latent variables

Hyperparameter	**Search Range / Options**	**Best value**
Hidden layer widths	Combinations from $\{12, 10, 8, 6, 4\}$ (strictly decreasing, depth ≤ 4)	$(12,\ 8,\ 6)$
L1 regularisation weight	$[10^{-8}, 10^{-5}]$ (log scale)	2.320×10^{-8}
L2 regularisation weight	$[10^{-6}, 10^{-4}]$ (log scale)	2.632×10^{-5}
Encoder activation	{ReLU, Sigmoid, Linear}	`linear`
Output activation	{ReLU, Sigmoid, Linear}	`sigmoid`
Latent activity L1	$[10^{-8}, 10^{-4}]$ (log scale)	1.232×10^{-8}
Learning rate	$[10^{-4}, 10^{-3}]$ (log scale)	8.976×10^{-4}
Batch size	$\{16, 32, 64\}$	16

The 3-latent search in Table 7.2 converges to a near-linear architecture with hidden widths $(12, 8, 6)$, a linear encoder, and a sigmoid output. Regularisation is dominated by L2 (2.632×10^{-5}), while both the weight L1 (2.320×10^{-8}) and latent activity L1 (1.232×10^{-8}) are effectively zero, indicating that all three latent coordinates are consistently utilised without requiring explicit sparsity pressure. The optimiser hyperparameters (learning rate 8.976×10^{-4}, batch size 16) remain in the same stable regime as in other runs. The resulting latent trajectories are shown in Figure 7.14.

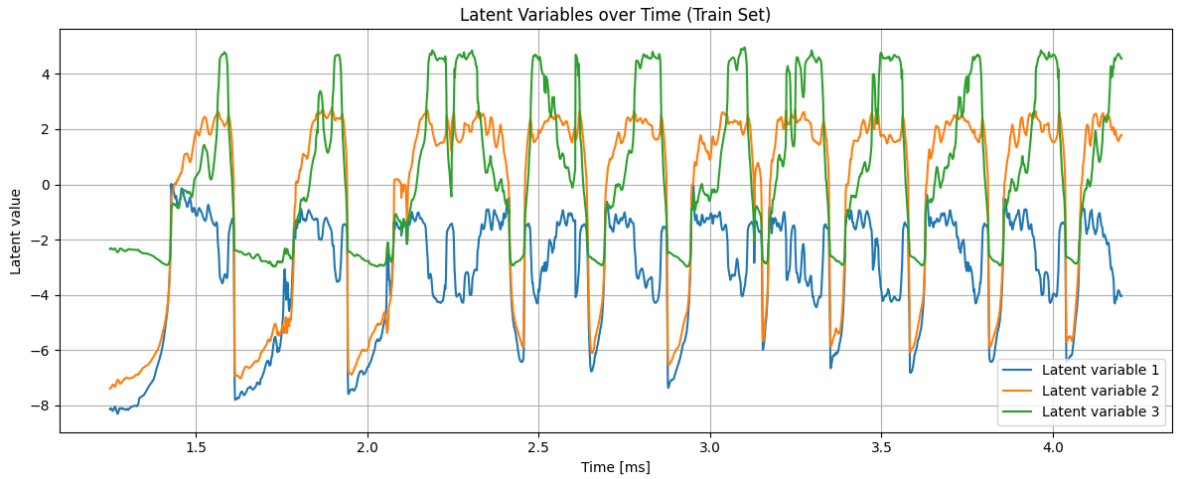


Figure 7.14: 3 Latent Variables Visualized

With three latents in Figure 7.14, the code remains cycle–locked but is very expressive. The second latent (orange) acts as a quasi-binary state variable, switching cleanly between active and relaxed phases. The first latent (blue) provides a continuous modulation within each cycle, tracking baseline level and slow intra-cycle morphology. The third latent (green) concentrates around the edges of the cycle: it rises sharply at onset and relaxes more gradually, capturing asymmetry between ignition/flashback entry and recovery, as well as medium-frequency structure that would otherwise be compressed. This division of roles—(i) state, (ii) envelope/shape, (iii) transition sharpness, yields trajectories that are stable across cycles with limited drift and only mild saturation at plateaus. Furthermore, the reconstructions of the thermodynamic and velocity features are shown in Figure 7.15.

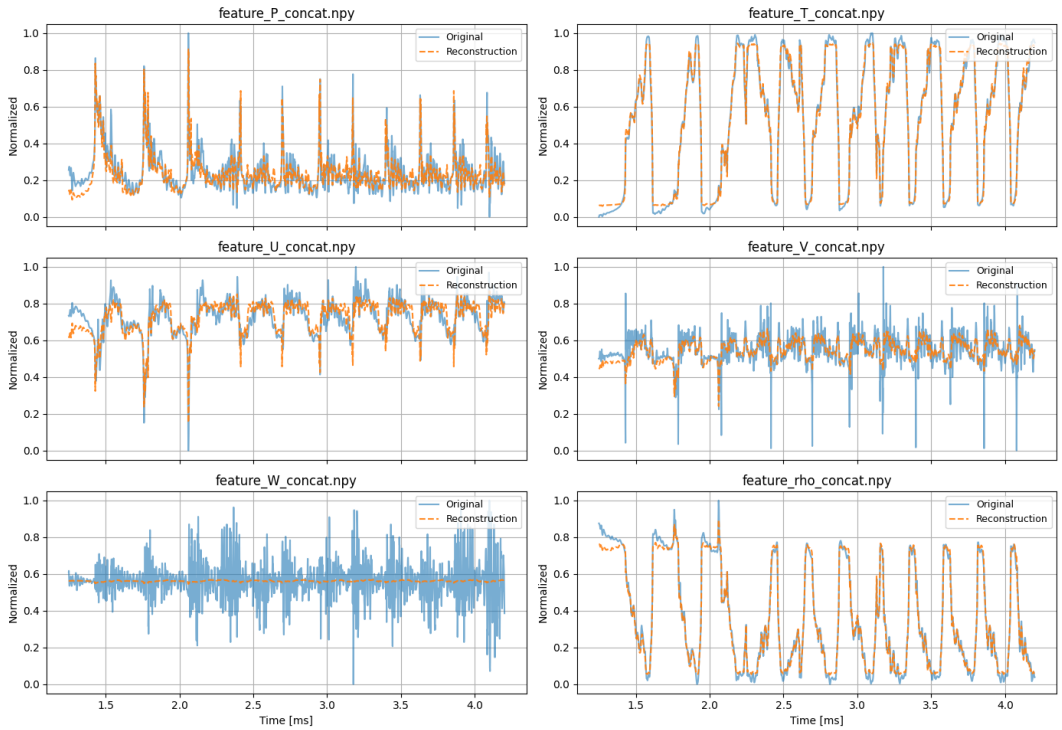


Figure 7.15: Reconstruction of Thermodynamic and Velocity Features with 3 Latent Variables

The reconstructions in Figure 7.15 remain excellent for the strongly cyclic, high–SNR variables (T and ρ), with only slight clipping at extrema. The noisier channels show clearer benefits from the three–dimensional bottleneck: for pressure P, the slow envelope and secondary modulations are tracked more faithfully, with reduced underestimation of intermediate peaks, while the sharpest spikes are still damped. The streamwise velocity u shows improved amplitude recovery and reduced phase lag around ramps, bringing the reconstruction closer to the baseline across cycles. The transverse velocities v and w exhibit better baseline alignment and less systematic bias; although high–frequency excursions remain smoothed, their magnitude is less underpredicted. In short, the third latent dimension is used to capture moderate–frequency structure in P, u, v, and w, while T and ρ are already reconstructed near ceiling fidelity. The latent trajectories in Figure 7.14 illustrate how distinct modes—state, envelope, and transition sharpness—map onto the observed improvements in these reconstructions. The corresponding species behaviour is shown in Figure 7.16.

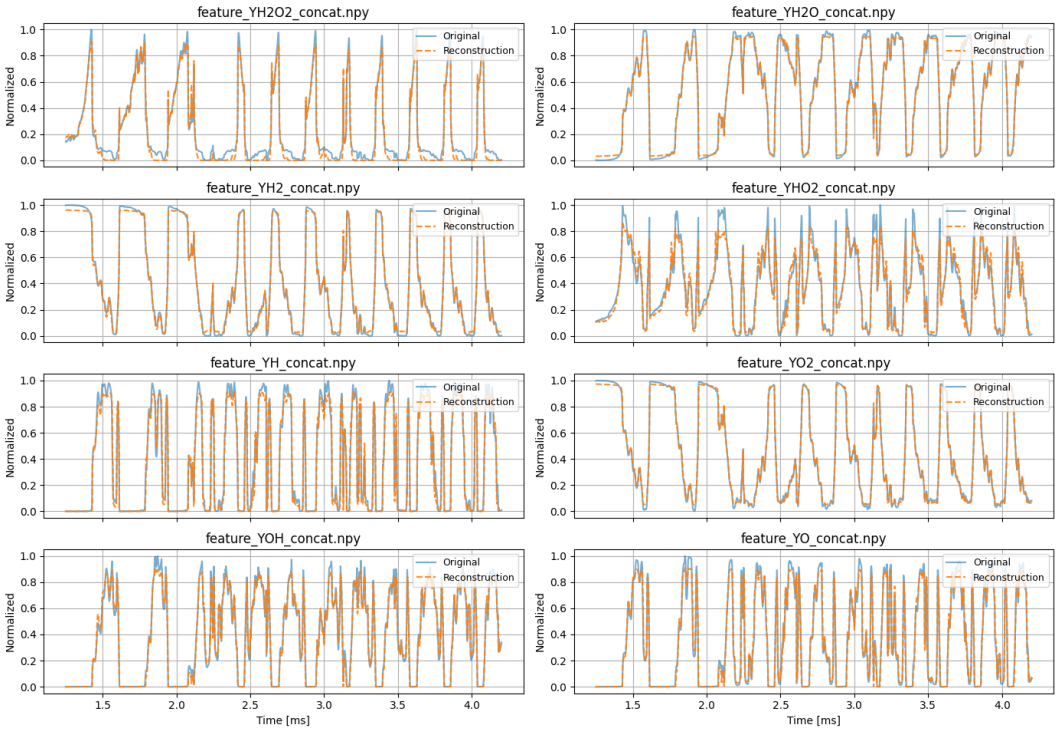


Figure 7.16: Testing of Mass Fraction Features with 3 Latent Variables

The reconstructions in Figure 7.16 are uniformly strong across species. Bulk species ($Y_{\mathrm{H_2O}}, Y_{\mathrm{H_2}}, Y_{\mathrm{O_2}}$) are essentially at ceiling, with waveforms and phase reproduced nearly perfectly and only marginal plateau bias. Clearer gains are visible in radicals and intermediates: $Y_{\mathrm{H}}, Y_{\mathrm{O}}, Y_{\mathrm{OH}}$ exhibit steeper ramps and better peak alignment, while $Y_{\mathrm{HO_2}}$ and $Y_{\mathrm{H_2O_2}}$ show sharper, less rounded pulses with improved timing. The multi–modal Y_{O} signal tracks intermediate maxima with reduced baseline drift. Residual discrepancies remain minor, including slight under/overshoot at the sharpest extrema (consistent with sigmoid saturation) and a small phase lag of only a few samples in the fastest transitions. The test reconstructions are considered next.

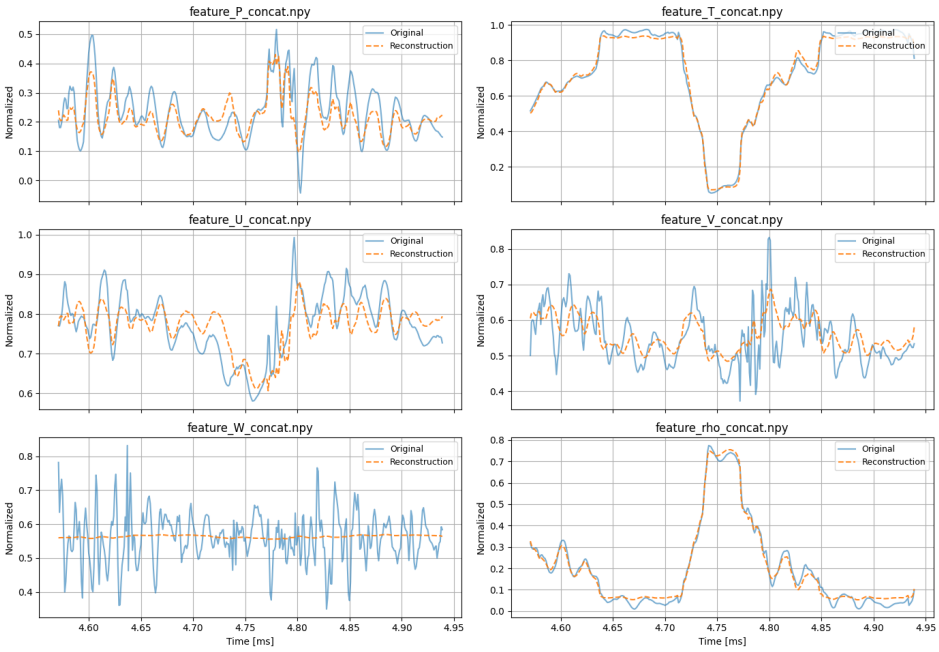


Figure 7.17: Testing of Thermodynamic and Velocity Features with 3 Latent Variables

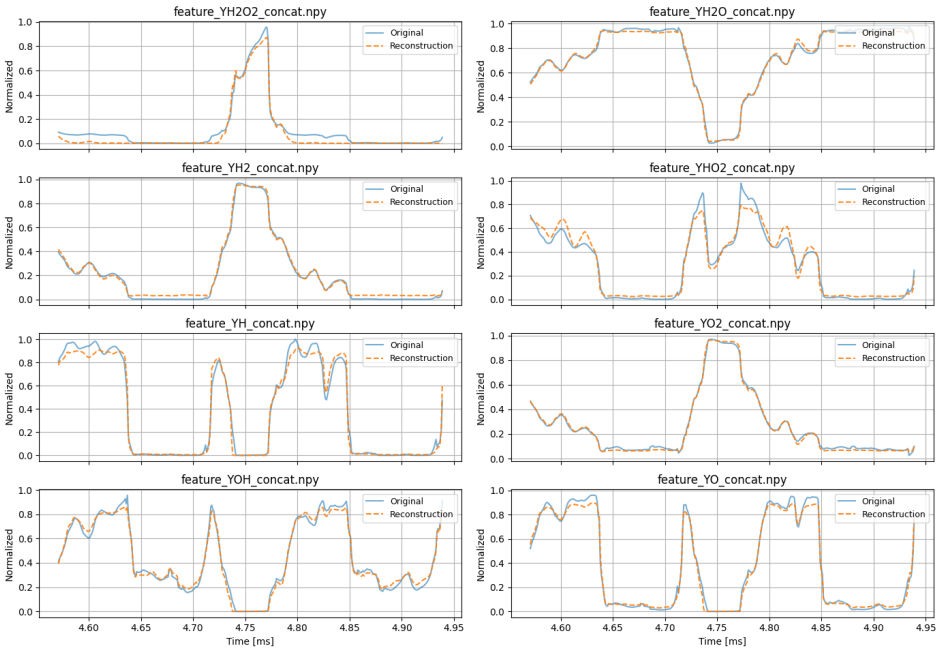


Figure 7.18: Testing of Mass Fraction Features with 3 Latent Variables

Relative to the training reconstructions in Figure 7.15, the test outputs in Figure 7.17 remain strong for T and ρ (near-identical phase and waveform, with only mild peak clipping). The noisier channels show the expected test–time smoothing: P, v, and w have slightly smaller amplitudes and occasional short lags at sharp ramps compared to train, while u retains accurate envelope tracking across cycles.

The species in Figure 7.18 generalise well and closely mirror their training counterparts in Figure 7.16.

Bulk species ($Y_{H_2O}, Y_{H_2}, Y_{O_2}$) remain essentially at ceiling on test; radicals/intermediates ($Y_H, Y_O, Y_{OH}, Y_{HO_2}, Y_{H_2O_2}$) show minor additional attenuation at sharp extrema relative to train but preserve the improved timing and peak matching. Residual discrepancies are limited to slight clipping near the most abrupt transitions, consistent with the sigmoid output nonlinearity.

Furthermore, the mean squared error distribution is as follows:

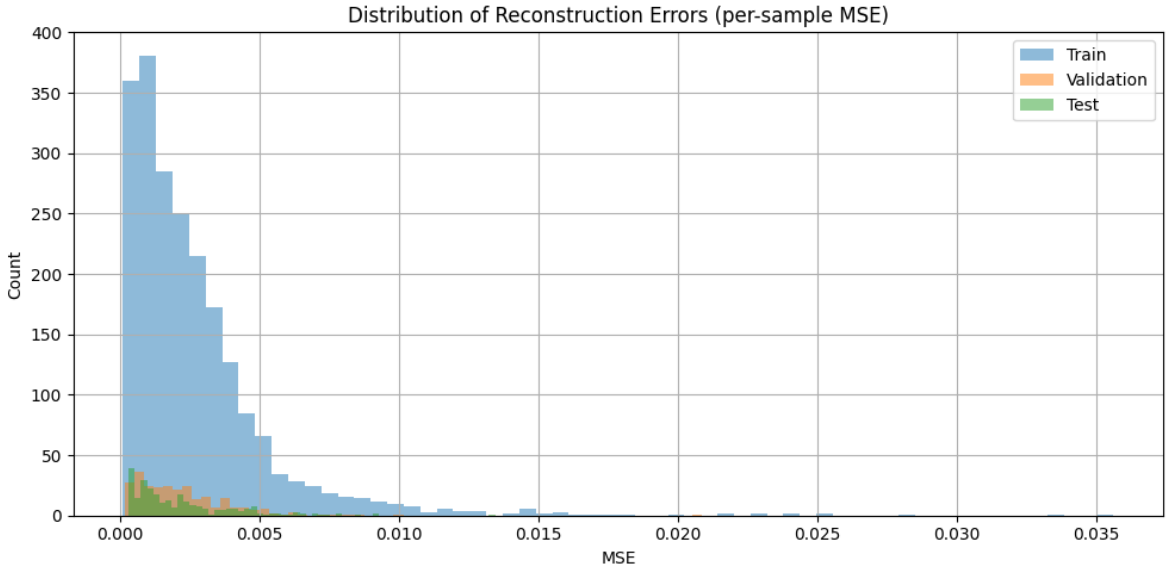


Figure 7.19: MSE Distribution for 3 Latent Variables

Figure 7.19 shows a right–skewed per–sample MSE distribution with most mass concentrated at very small errors ($\lesssim 3\times10^{-3}$) and a curtailed long tail, indicating that only a small fraction of samples are challenging to reconstruct. Variance is reduced across all splits, with the tightest spread on validation/test. The ordering train $>$ val $>$ test persists, which points to cleaner held–out segments rather than overfitting. Overall, the three–latent model achieves both low central tendency and dispersion of error, with tail events largely confined to sharper transients (e.g. in P, u, v, w) and the more structured species. Finally, the loss is shown:

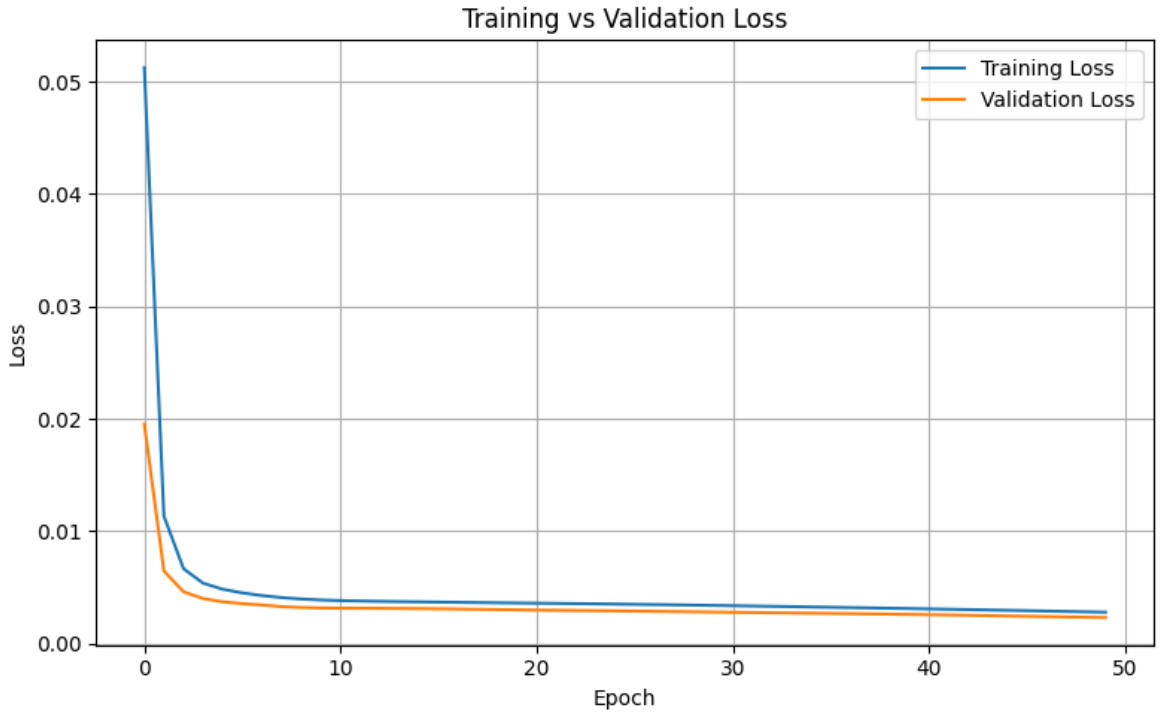


Figure 7.20: Loss Evolution for 3 Latent Variables

The training/validation curves in Figure 7.20 show a rapid drop over the first $\sim$3 epochs, followed by a smooth, monotonic decline and a long, shallow tail. The validation curve remains slightly below the training curve throughout, never turning upward, which indicates an absence of overfitting and, as before, suggests that the training segment is intrinsically harder/noisier than validation. The overall behaviour corroborates the reconstruction and MSE results that having 3 latent variables lowers both the central tendency and dispersion of the error without inducing validation drift.

7.2.2. Summary

Across the three autoencoders (2, 3, and 4 latents), reconstruction quality improves monotonically with latent dimension. The largest gain occurs from 2$\rightarrow$3 latents; adding a fourth latent yields a consistent but smaller refinement. Variables with strong, low–frequency cyclicity (T and ρ) and the bulk species (H_2O, H_2, O_2) are reconstructed almost perfectly in all settings. The noisier/less regular channels (P, u, v, w) and the more structured species (e.g. H, O, OH, HO_2, H_2O_2) benefit most from the added capacity: amplitude bias decreases, mid–frequency content is better captured, and small phase lags are reduced. Per–sample MSE histograms are right–skewed but shift left and tighten from 2$\rightarrow$3$\rightarrow$4 latents, indicating both fewer high–error outliers and lower central tendency. Loss trajectories are smooth and monotone with validation below training throughout, consistent with a chronological split in which the held–out segments are slightly cleaner; there is no evidence of overfitting.

A 3–dimensional bottleneck is a strong operating point: it captures nearly all recoverable structure at modest complexity, while a 4–dimensional code mainly polishes transition dynamics (diminishing returns). For downstream clustering, the latent space should, by design, separate state from noise, so it is expected to have cleaner group structure when clustering $\mathbf{z}_t$ (or short time–window summaries of $\mathbf{z}_t$) rather than the raw features. With a 2 latent code, one coordinate typically behaves like a state/phase indicator and the other as a continuous intensity; therefore, two compact clusters (relaxed vs. active) are expected with transition samples forming a narrow bridge or fuzzy boundary between them. Moving

to 3 latents should make the transition dynamics more explicitly represented (e.g., onset/offset sharpness), so a separate “precursor/transition” cluster is more likely to emerge and cluster quality metrics (e.g., silhouette) would be expected to improve. A 4 latent code would mainly refine this picture: a substructure within the active state (early/late active, strong/weak cycles) and slightly cleaner boundaries would be anticipated, but with diminishing returns and a higher risk that very brief events form tiny fragments if the clustering method is too aggressive. In conclusion, the three-latent configuration is adopted for all subsequent clustering analyses, as it achieves the best balance of accuracy and interpretability with minimal complexity.

7.2.3. Robustness

As a robustness test of unseen data, a step is taken past the test section, where data from S4 is sampled. These datapoints are run through the autoencoder’s best pretrained configuation; the 3 latent variable structure. The final encoder output is shown in Figure 7.21,

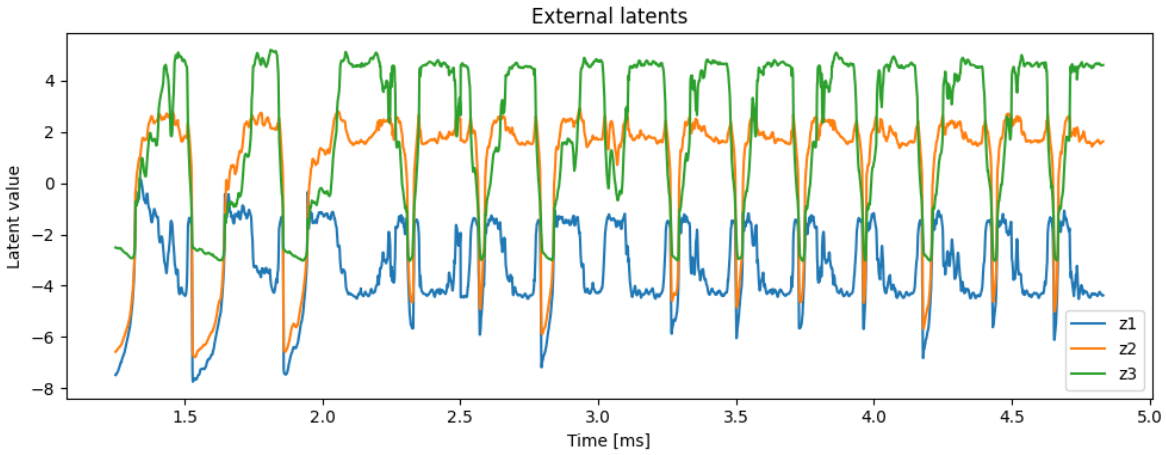


Figure 7.21: Latent Variables of the S4 Sampling Point

where the latent variables qualitatively resemble Figure 7.14, regarding high and low frequency nodes such as noise and cycles, respectively. The reconstruction of the decoder is then plotted against the original S4 data, which is shown in Figure 7.22 and Figure 7.23.

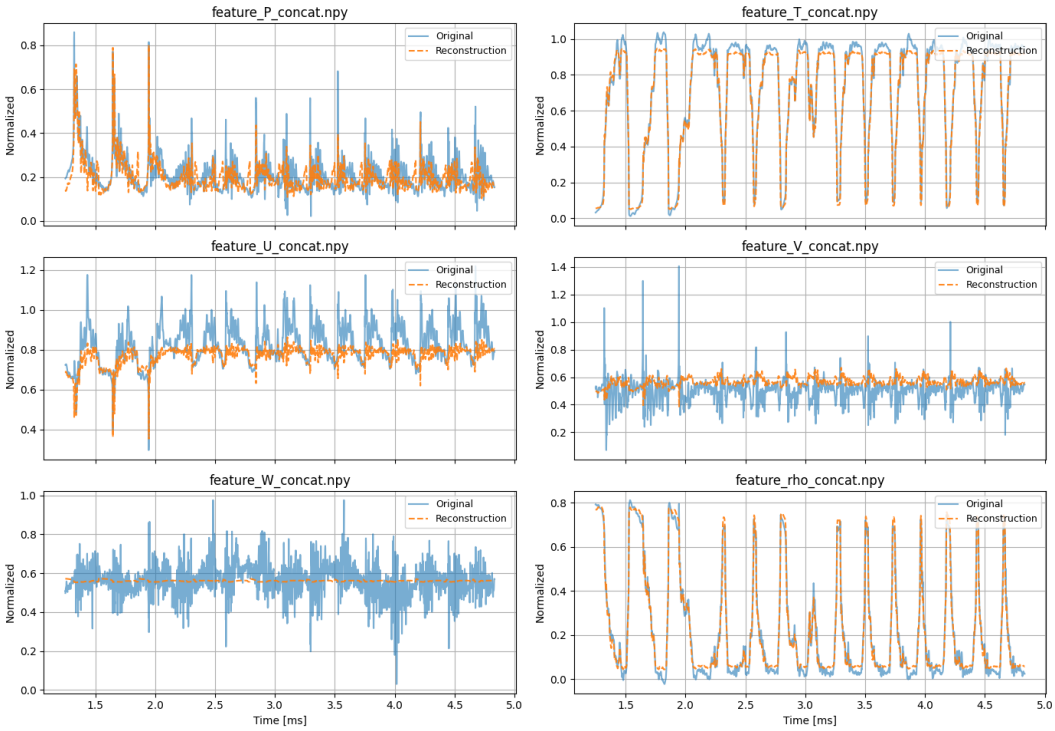


Figure 7.22: Reconstruction of Thermodynamic and Velocity Features of S4

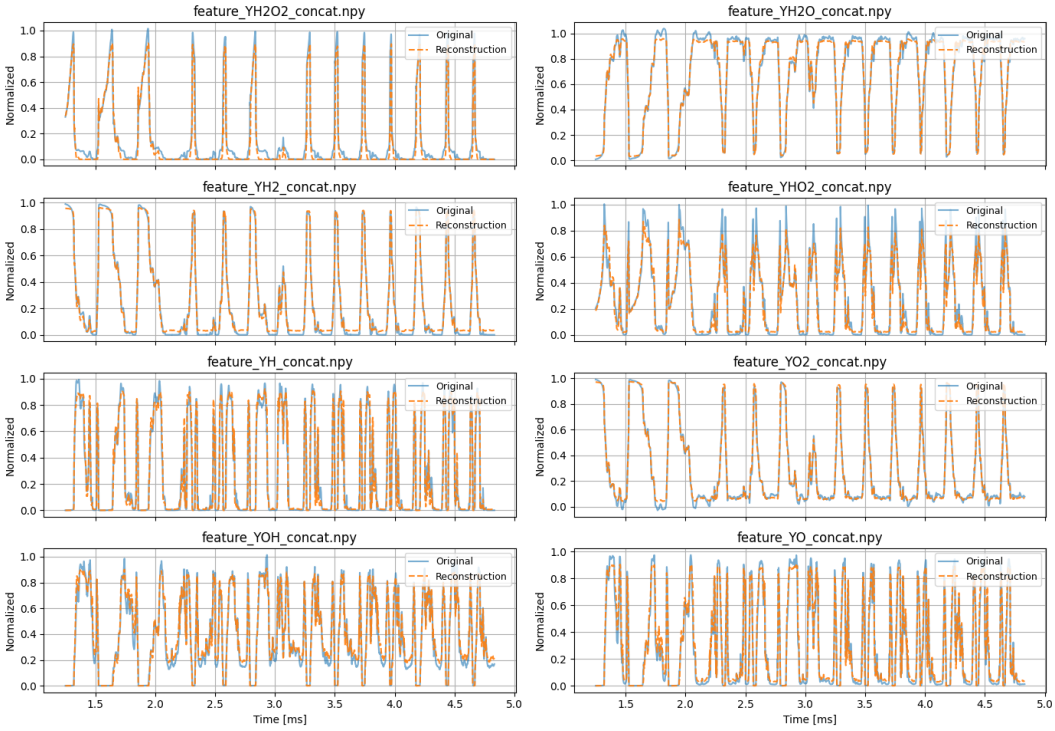


Figure 7.23: Reconstruction of Mass Fraction Features of S4

A comparison of the reconstruction input overlays for the S4 data in Figure 7.22 against those from

the Figure 7.15 S3 training reveals clear variable-dependent generalization. For the thermo-chemical variables T and ρ, the external reconstructions are high-fidelity: the periodic depressions, recovery phases, and pulse shapes are reproduced with only minor peak clipping. In contrast, P, U, V, and W degrade highly at S4: the decoder exhibits variance shrinkage (from the amplitude compression), and suppression of sharp excursions, so bursts and troughs visible in the inputs are flattened in the reconstructions. At S3, these same variables are tracked more closely, with moderate smoothing but substantially better amplitude agreement.

Furthermore, a side-by-side inspection of the species mass–fraction reconstructions in Figure 7.23 and Figure 7.16 shows that the autoencoder transfers remarkably well from the training location to S4. For all plotted channels ($Y_{\mathrm{H}}, Y_{\mathrm{H_2}}, Y_{\mathrm{H_2O}}, Y_{\mathrm{H_2O_2}}, Y_{\mathrm{HO_2}}, Y_{\mathrm{O}}, Y_{\mathrm{O_2}}, Y_{\mathrm{OH}}$), S3 exhibits near-perfect overlap between input and reconstruction, with only minor smoothing at sharp corners. At S4, the cycles remain highly faithful; periods, smaller cycles, and decay tails are preserved. However, small, systematic deviations appear: peaks are slightly attenuated and occasionally clipped (most visible for radical-rich channels $Y_{\mathrm{HO_2}}, Y_{\mathrm{OH}}, Y_{\mathrm{O}}$), and some rising edges show a subtle temporal lag relative to the input. Moreover, stable bulk species such as $Y_{\mathrm{H_2}}$ and $Y_{\mathrm{O_2}}$ are reproduced almost indistinguishably from the truth. These patterns suggest that the learned thermo-chemical features generalizes well across locations; the residual errors are characteristic of mild low-pass smoothing and output saturation near extrema, likely arising from distribution shift outside the training min–max range and the decoder's sigmoid output. Overall, the autoencoder captures species dynamics robustly at the unseen location, with only modest peak underestimation concentrated in fast, radical transients. The general conclusion might be that the noisy features may be under-representated while the low frequency modes match very well for S4. This is visible in Figure 7.24,

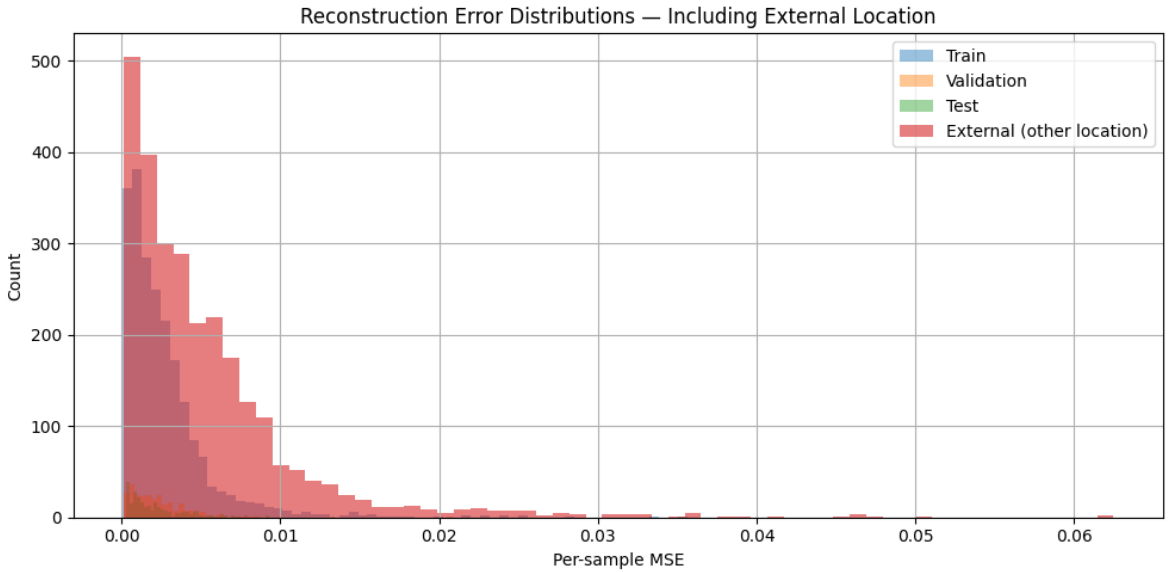


Figure 7.24: MSE for S4

where the Figure 7.24 shows the distribution of the MSE of S4 overlayed on the previous Figure 7.19. S4 (red) remains clearly shifted to the right relative to the train/validation/test splits and exhibits a heavier upper tail (extending to roughly 6×10^{-2}), indicating both more frequent and larger errors at the unseen location. Nevertheless, there is substantial overlap with the in-split distributions near zero, showing that many external time steps are still reconstructed well while a smaller fraction accounts for the long tail of larger errors. Bar heights reflect sample counts and depend on sequence length and binning;

the primary comparative signals are the lateral shift of the external distribution and its pronounced right tail.

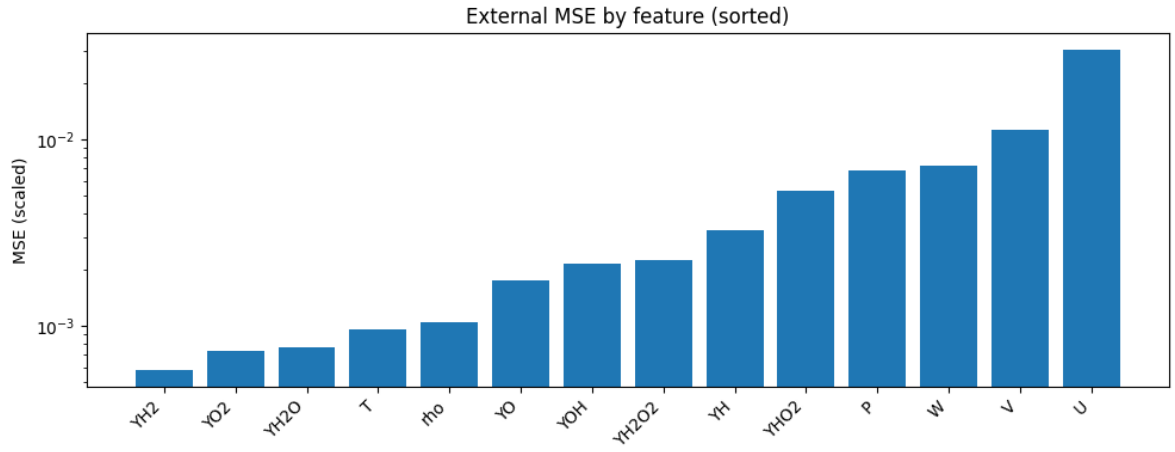


Figure 7.25: MSE by Feature

Furthermore, Figure 7.25 summarizes the MSE with a logarithmic y–axis. Errors are smallest for thermo–chemical variables such as $Y_{\mathrm{H_2}}$, $Y_{\mathrm{O_2}}$, $Y_{\mathrm{H_2O}}$, and remain low for T and ρ, indicating close form agreement. Intermediate errors are observed for radical–rich species (Y_{O}, Y_{OH}, $Y_{\mathrm{H_2O_2}}$, Y_{H}), while the other channels U, V, W, and P dominate the error budget by over an order of magnitude. Because values are scaled and the decoder output is bounded, excursions beyond the training range and mild peak clipping can inflate MSE; nonetheless, the ranking clearly localizes where generalization is weakest (primarily the velocity/pressure signals). Finally, the scatter plot in Figure 7.26 is obtained by plotting the first two latent variables, similar to a phase diagram. Each dot is one time sample. The blue cloud delineates the S3 data, while the orange cloud shows how S4 maps into the same space. Substantial overlap indicates that the external inputs are encoded near the training data and are thus easier to reconstruct; systematic offsets or shape differences (”latent drift”) indicate distribution shift, where the decoder must extrapolate and reconstruction error typically grows. In the illustrated case, the external points partly overlap the training manifold but deviate along specific regions, consistent with the observed degradation for some of the velocity and pressure channels.

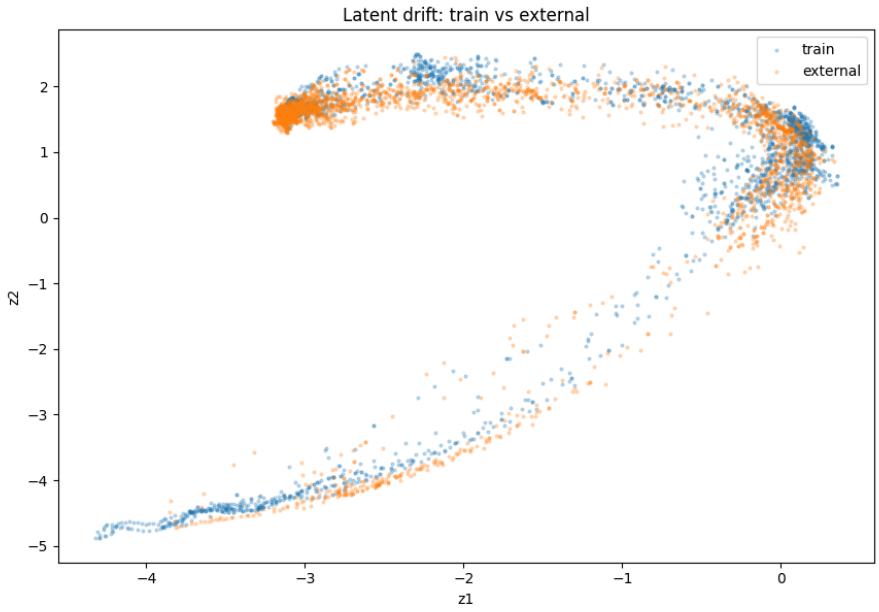


Figure 7.26: Latent drift for S3 (blue) and S4 (orange)

Now, the precursor detection can be discussed.

7.3. Precursor Detection

This section presents the results of the precursor detection analysis. As introduced in Section 6.2, the modularity-based clustering algorithm operates on the latent variables, representing them in a phase space diagram along with a probability transition matrix. These components are then used to construct a weighted graph, which serves as the basis for clustering. Using a modularity metric, the network is partitioned into clusters, with the goal of categorizing the time series into normal, precursor, and extreme states.

The section begins with the best-case configuration, using data from S3 and three latent variables. The process described above is detailed step by step. Following this, a robustness analysis is carried out, exploring the impact of varying latent variable structures, extreme-value thresholds, and clustering on unseen data. Finally, the original features employed by Floris [42] are incorporated to highlight how the present work builds upon prior research.

7.3.1. Tessellation and Weighted Graph

Initially, the latent variables are normalized, such that one does not have more influence than the other.

To begin, a latent variable is chosen to bear the extreme threshold. This variable is chosen as the second variable in Figure 7.14 referred to as L1, due to its regular and cyclic nature. Moreover, it resembles the temperature plot very closely, a feature used in [42]. However, the extreme value that is set is more arbitrary, due to the normalized values of the latent variables. The best configuration has been found using an extreme value of 0.25, shown in Figure 7.27.

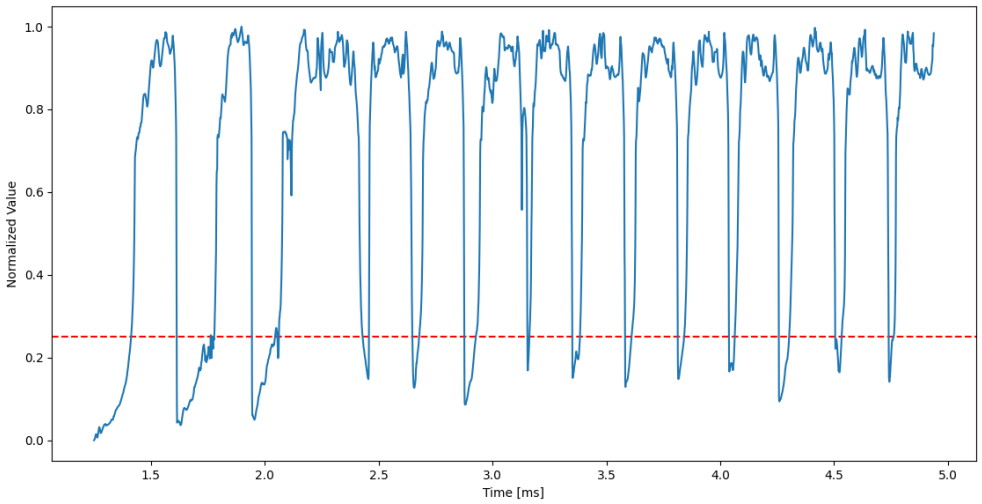


Figure 7.27: Extreme Threshold on Chosen Latent Variable

This extreme value has little effect on the precursor time, as will be discussed in the robustness. The time series is also split, where the first part is used in this section, and the following part in the robustness. Furthermore, the phase space of the time series is shown in Figure 7.28, where L1 is plotted against the first latent variable, L0, where L1 and L0 are both normalized using min-max.

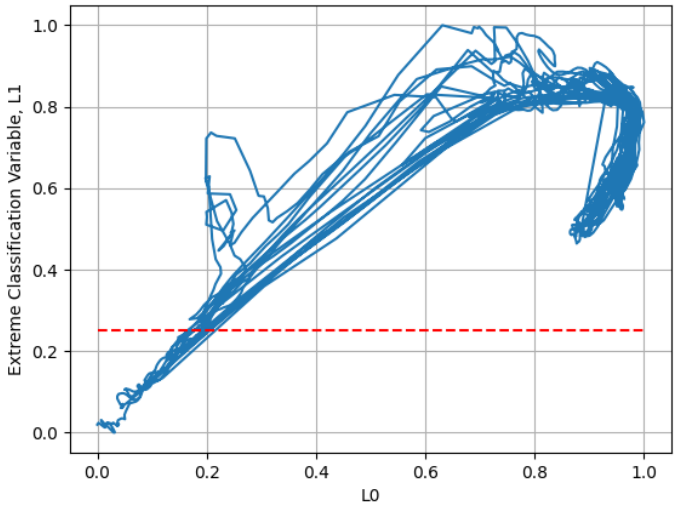


Figure 7.28: Phase Space Diagram of L1 against L0, Extreme=0.25

After this, the diagram is tessellated. The extreme regions are identified and stored separately in order to initialize the clustering algorithm with two distinct communities: an extreme community and a non-extreme one, a modification to the algorithm of Golyska and Doan [46] by Floris [42] where any community above the threshold is immediately separated as extreme. The tessellation is shown below.

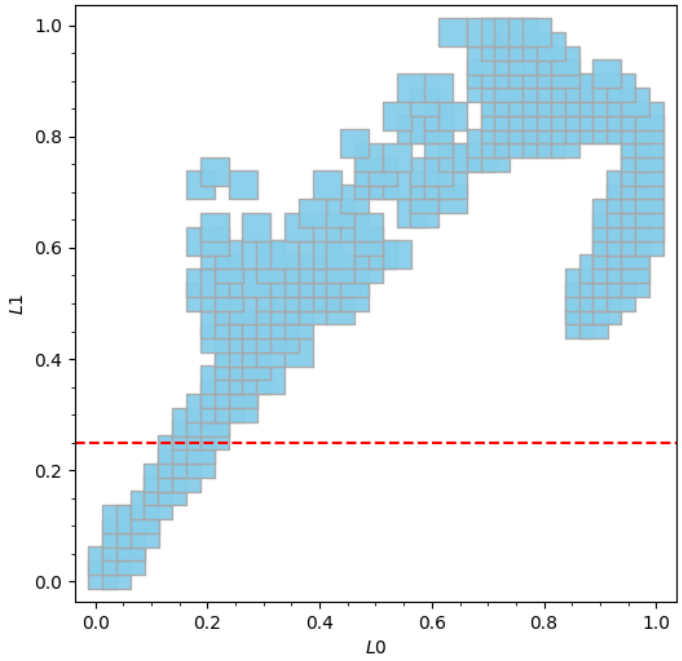


Figure 7.29: Tessellation of L1 against L0

In this diagram, the chosen tessellation size, $M = 30$, reflects a balance between computational efficiency and the accuracy with which the tessellated phase space represents the underlying trajectory. As noted in the analysis by Golyska and Doan [46], too few tessellation sections fail to capture the system dynamics in sufficient detail, making it difficult to distinguish between normal and precursor clusters. On the other hand, increasing M excessively results in significantly higher computational costs, scaling with M^{N_f}, where N_f is the number of features; without improving prediction performance.

In the subsequent stage, the algorithm proceeds into an iterative loop. The system is initially expressed as a transition probability matrix, which is then converted into a graph and partitioned using the modularity-based clustering approach. The resulting reduced graph is subsequently mapped back into a transition probability matrix, and this process is repeated. At the conclusion of each iteration, the transition probability matrix together with the reduced graph defines the current state of the system, as illustrated in Figure 7.30 and Figure 7.31.

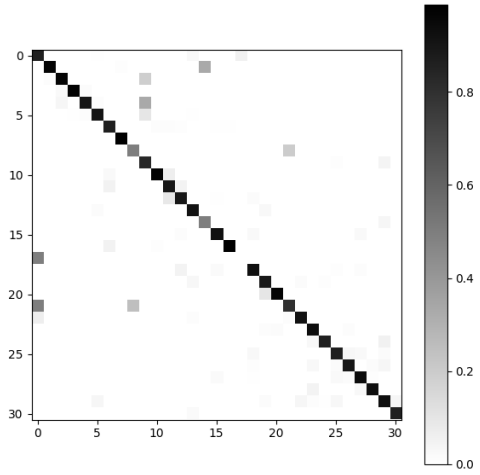


Figure 7.30: Transition Probability Matrix

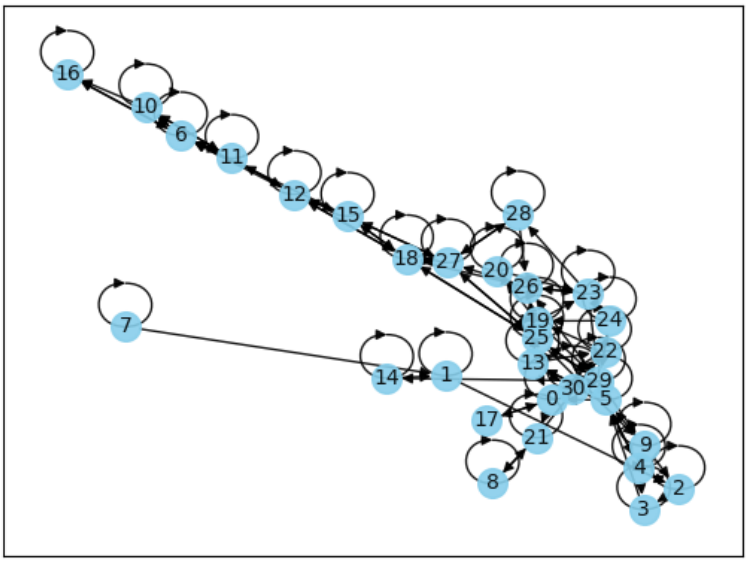
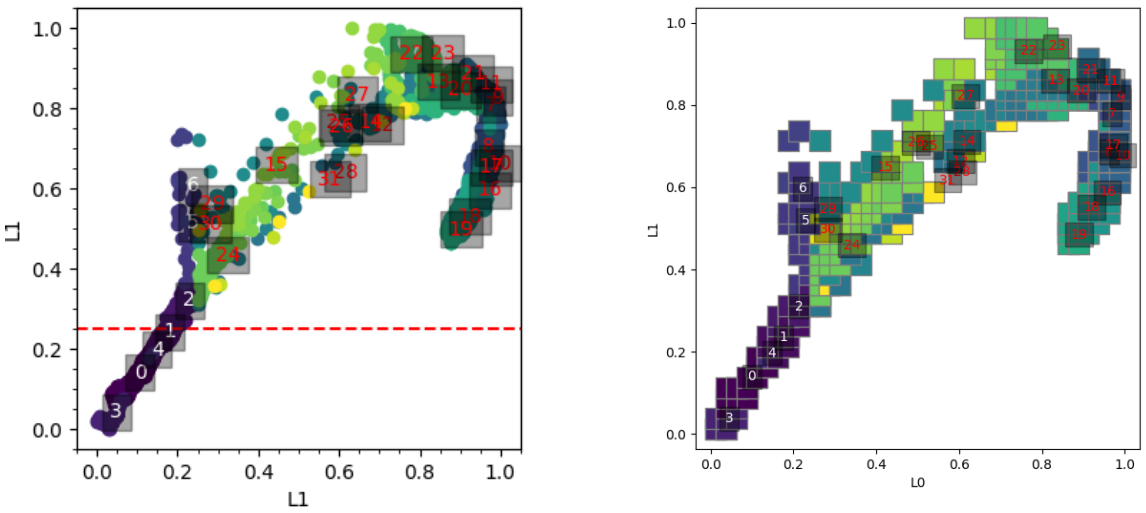


Figure 7.31: Weighted Directed Graph

The loop continues until one of two termination criteria is met: either the maximum number of iterations is reached, or the number of clusters decreases below a prescribed threshold. The maximum number of iterations is fixed at 10, as beyond this point no significant structural changes in the system are observed. The minimum number of clusters is set to 15 in order to preserve a sufficiently detailed representation of the system's dynamics while maintaining computational tractability.

7.3.2. First Clustering

Furthermore, the identified clusters are shown in the phase space diagram and tessellated graph in



(a) Clustered trajectory in phase space

(b) Clustered tessellated space

Figure 7.32: Visualization of clustering applied to the system: (a) trajectory in phase space and (b) corresponding tessellated representation.

Based on the graphs above, the clusters in Figure 7.33 are found,

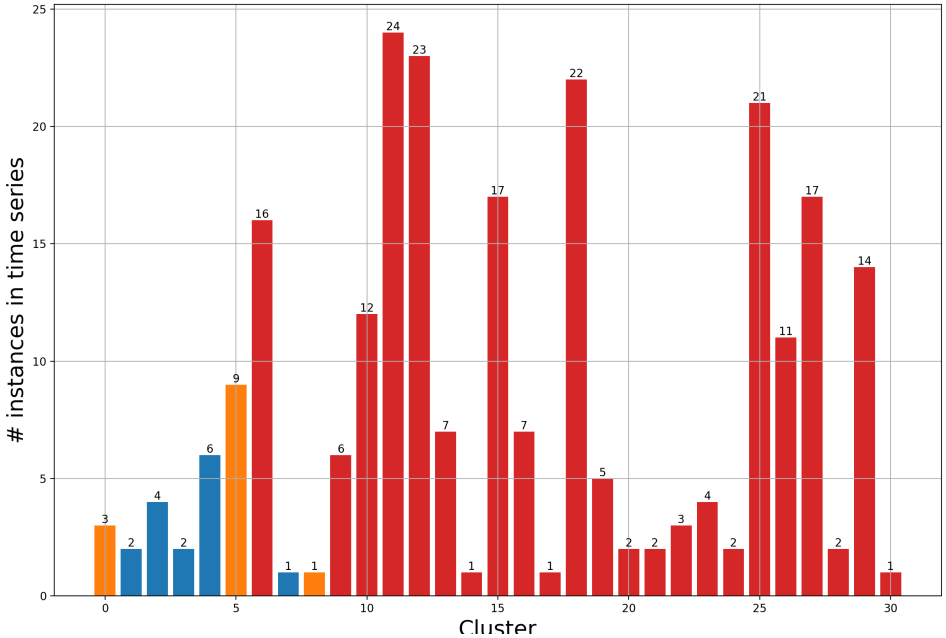


Figure 7.33: Cluster Types

where the red, orange, and blue colors correspond to the extreme, precursor, and normal clusters, respectively. The plot represents the number of hypercubes assigned to each cluster or community. As shown, clusters 2, 3, 4, 5, and 8 correspond to normal behavior, while clusters 1, 5, and 8 are identified as precursors; the remaining clusters are classified as extreme. In general, the more regular

a time series is, the fewer clusters are required in each category. This is because a narrow value range is typically represented by a single cluster, whereas if the precursor states span a broader range of values, the formation of multiple clusters would be necessary to capture their variability.

7.3.3. Time Series Result

Finally, based on these clusters, the affiliation can be created between the clusters and indices, forming the final partition of the time series, found in Figure 7.34

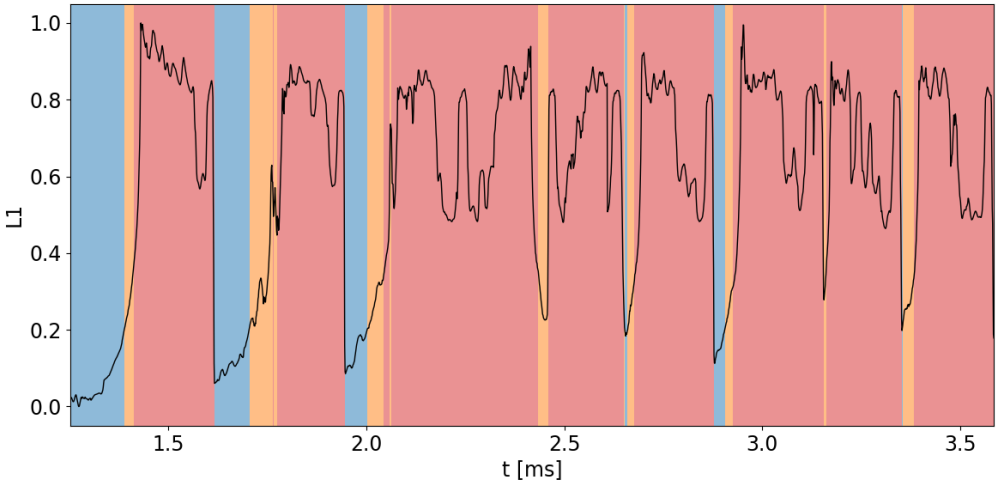


Figure 7.34: Fully Clustered L1

From the clustering results, the temporal evolution of the combustor can also be interpreted directly from the temperature signal. Figure 7.34 shows the normalized latent variable evolution sampled over time, with the background color denoting the cluster type: blue corresponds to normal operation, orange to precursor states, and red to extreme states.

It can be observed that prior to each flashback event, the algorithm consistently identifies a precursor state (orange) before the system transitions into the extreme regime (red) without any false negatives (FN), with one small false positive at 2.7 ms, occuring right after an extreme event. This general behavior demonstrates the ability of the method to provide a predictive warning ahead of the rapid rise in temperature associated with flashback. In addition, the time intervals spent in the normal regime reveal the strongly unstable character of the combustor: the system frequently departs from the normal state into precursor or extreme conditions.

The clustering is therefore not based solely on instantaneous values of the latent variables, but rather on the trajectory of the system through the phase space. This is evidenced by the classification of short segments into distinct clusters even when their latent values overlap. As a result, the approach highlights the dynamical precursors of flashback events and underscores its potential as an early-warning diagnostic tool for unstable combustion systems. The exact indices are reflected onto the temperature plot in Figure 7.35.

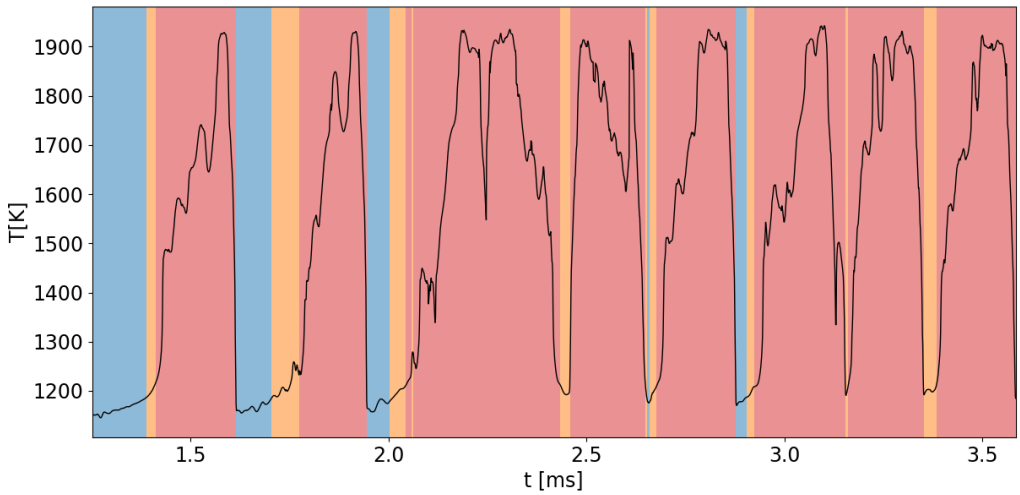


Figure 7.35: Clusters Mapped onto Temperature

As observed, the similarity between one of the latent variables and the temperature feature results in a one-to-one mapping of the extreme and precursor zones. The maximum warning time identified is approximately 42 μs, which corresponds to a temperature difference exceeding 100 K, highlighting the importance of such a prediction horizon. Moreover, even when extreme events occur in close succession, leaving only a short relaxation period, the algorithm is still able to distinguish between normal and precursor clusters. Next, the robustness analysis will be performed.

7.3.4. Robustness Analysis

This subsection will discuss all the robustness analysis performed on the precursor detection.

Latent Variable Structure

For the robustness study, clustering is carried out in the latent space learned by the autoencoder. Although the three-dimensional embedding yielded the strongest reconstruction performance, the clustering algorithm is also applied to embeddings with $d \in \{2, 4\}$ latent dimensions to assess sensitivity to representation size, specifically, whether clustering benefits from more or less information, or whether the $d = 3$ configuration remains optimal as found before. For each case, one latent variable was found that corresponded mostly to the cyclic variations of features like temperature. This latent variable was chosen to set the threshold of extreme. Firstly, the result of the two latent structure is shown,

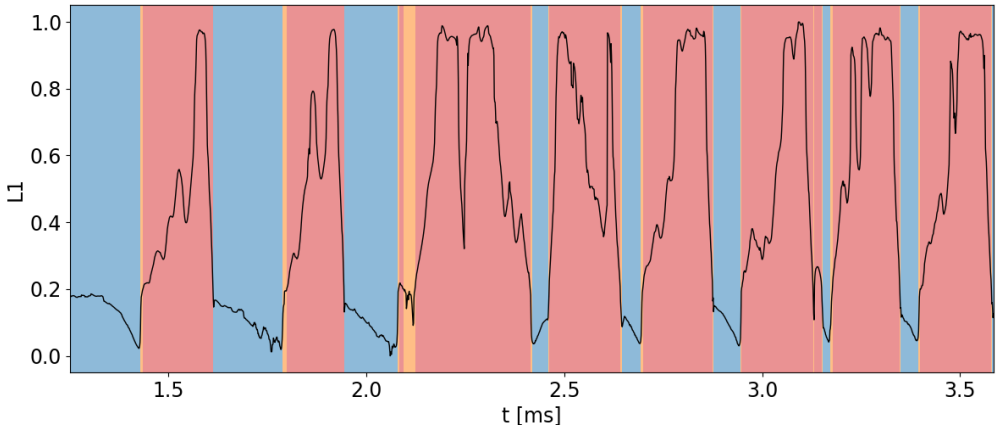


Figure 7.36: Clustering Analysis for 2 Latent Variables

Relative to the 3 latent configuration in Figure 7.34, the 2 latent case produces a noticeably more fragmented segmentation (many short orange slivers) and several precursor intervals that do not terminate in an event (FP). In both configurations the extreme events are consistently preceded by a precursor (no false negatives are evident). However, the 3 latent model yields cleaner, longer, and more stable precursor windows immediately upstream of each event, whereas the 2 latent model tends to compress the precursor and insert false ones. Hence, with comparable absence of false negatives but a lower false–positive rate and much shorter warning lead times, the 3 latent configuration provides the more reliable early–warning signal. Furthermore, the 4 latent structure is shown in Figure 7.37

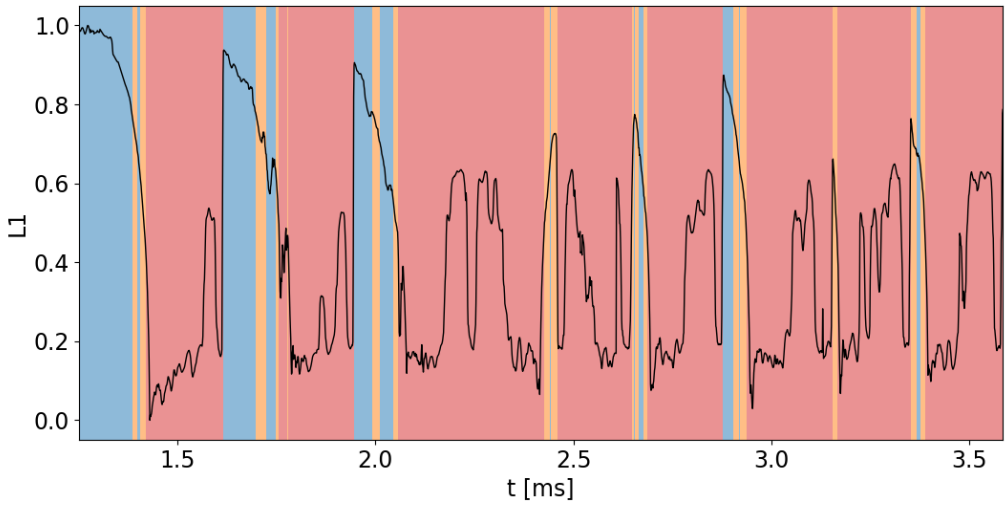


Figure 7.37: Clustering Analysis for 4 Latent Variables

Using four latent variables yields a more reactive, but less precise—segmentation of the latent variable. It should be noted that the latent variable corresponding to the temperature feature, in this structure, was upside down. Extreme events remain consistently preceded by a precursor (no false negatives), and the onset of each precursor is essentially unchanged relative to the 3 latent configuration, indicating that early-warning lead time does not improve with $d = 4$. The principal difference is additional false positives (30 total) and unstable precursor duration; the trailing boundary between precursor and event

toggles frequently, while the first index of the precursor is stable. Consequently, the 4 latent model offers no gain in warning time and degrades precision, whereas the 3 latent configuration provides cleaner, longer precursor windows with fewer false activations and is therefore preferred. It should be noted that the clustering time was two orders of magnitude longer for the 4 latent model; an additional disadvantage. It is concluded that a 3 latent variable model is indeed better for both the autoencoder, and the clustering algorithm. Next, the extreme value is varied.

Extreme Value Threshold

Secondly, the extreme value threshold was varied, to see whether a small increment can drastically change the precursor time, or the false positive rate. For all values attempted, the false negatives were zero, indicating there was no presence of an extreme event without a precursor, however false positives did happen. For this analysis, three flashbacks were observed with long precursors, indicating that they should be more sensitive to extreme value changes. These flashbacks correspond to times 1.5 ms, 1.85 ms, and 3 ms in Figure 7.34. Their precursor duration is shown below,

Table 7.3: Precursor intervals identified at different extreme value thresholds.

Threshold	Precursor 1 [ms]	Precursor 2 [ms]	Precursor 3 [ms]	False Positives
0.28	1.407–1.420	1.720–1.749	2.911–2.942	3
0.27	1.386–1.413	1.704–1.749	2.903–2.940	4
0.26	1.387–1.412	1.704–1.749	2.904–2.930	1
0.25	1.388–1.412	1.704–1.746	2.906–2.925	1
0.24	1.386–1.412	1.704–1.746	2.905–2.925	2
0.23	1.388–1.406	1.704–1.741	2.906–2.922	2
0.22	1.388–1.406	1.704–1.741	2.906–2.921	3

The principal outcome of the threshold ranges in Table 7.3 is that the onset of the precursor is essentially invariant to the extreme–value threshold. Across thresholds $0.27 \rightarrow 0.22$, the first detected time of Precursor 1 stays within 1.386–1.388 ms (a spread of ≤ 0.002 ms), Precursor 2 is fixed at 1.704 ms, and Precursor 3 lies in 2.903–2.906 ms (spread ≤ 0.003 ms). Only the most stringent case (0.28) delays the onset slightly (by 0.015–0.021 ms), as expected from a harder threshold. By contrast, the trailing boundary between precursor and the extreme event shifts by up to 0.014–0.021 ms as the threshold is varied, which merely changes where the thresholded signal crosses into the event; it does not affect the early-warning lead time. In other words, the detector consistently captures the very first departure from nominal behavior, while threshold changes mostly trim the end of the precursor segment. Practically, a mid-range choice (e.g. 0.25–0.26) yields only one false positives in these tests while preserving the same invariant onset, therefore, 0.25 is adopted as the default in subsequent analyses. Next, the unseen data is clustered.

Unseen Data

This section examines the robustness of the algorithm with respect to the available time–series length, serving also as a first step toward online deployment. Because updating clusters in real time is computationally prohibitive, performance is assessed in a ”train-on-prefix, predict-on-suffix” protocol: clusters are learned from an initial segment of the data, and the remaining samples are assigned without reclustering. Each new point is labelled as normal, precursor, or extreme according to the distance to the nearest precomputed cluster, and the prediction time is evaluated exactly as in the baseline analysis.

Concretely, out of a 3.7 ms record (4500 time steps), the first 2.3 ms (2330 steps) are used to form clusters in the latent variables. The suffix is then classified by nearest–cluster assignment, and the resulting continuation of the latent variable is shown in Figure 7.38.

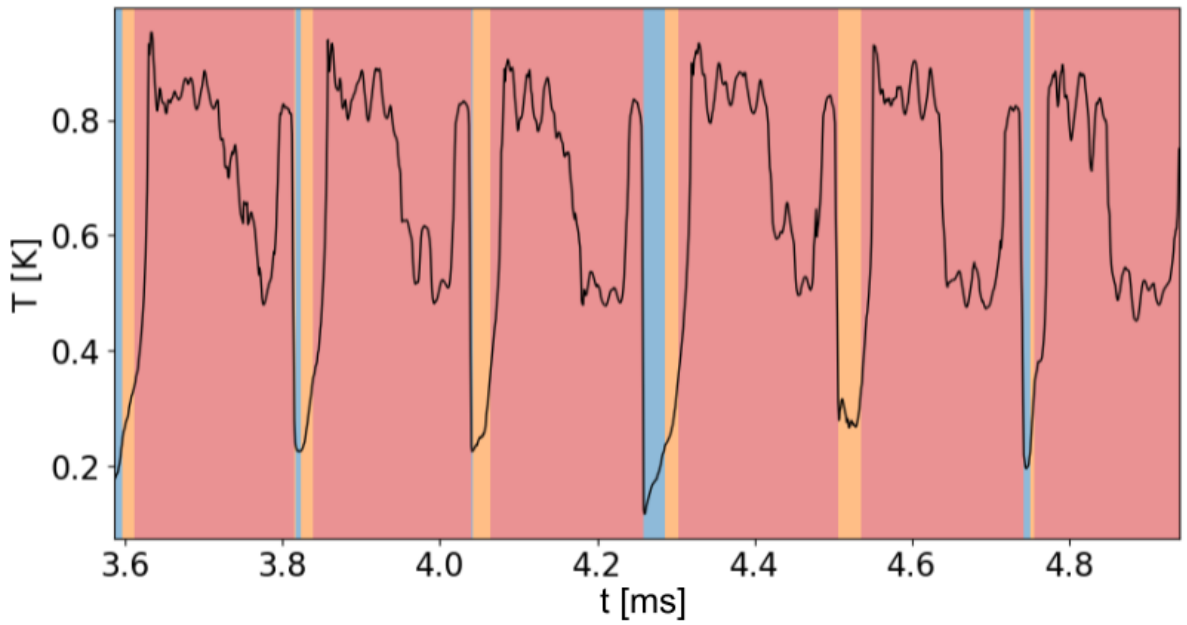


Figure 7.38: Nearest Cluster Analysis for Unseen Data

Figure 7.38 presents six flashback episodes classified by mapping each sample to the nearest precomputed cluster. The regime labelling is coherent across all events: every flashback is preceded by a precursor interval, no missed precursors are observed, and nominal segments appear only between events. In five of the six cases the standard sequence normal to precursor to extreme is recovered. The fourth episode exhibits essentially no normal relaxation and maintains an elevated relaxation temperature; consequently, no nominal label is assigned, an outcome that is plausibly consistent with the underlying physics.

A single weakness is a very short ($\approx 2\ \mu$s) precursor appended to the end of the first flashback. In the remaining episodes the termination of the extreme phase coincides with an abrupt drop (large negative temperature gradient), whereas in the first episode the relaxation is more gradual; the time-invariant, distance-based classifier thus carves out a small precursor-like sliver at the tail of the event. Precursor durations are shorter on average ($\approx 23\ \mu$s), largely reflecting the generally shorter relaxation intervals in this late-time segment and the single short false positive. Overall, the fixed clusters transfer well to unseen sequences: regime transitions are identified consistently and the early-warning signal is preserved.

7.3.5. S4 Clustering

As a last robustness, the latent variables in Figure 7.21 produced from the robustness section of the autoencoder, subsection 7.2.3, will be given to the clustering algorithm. Again, the latent variable corresponding to a cylic temperature feature is used as a control.

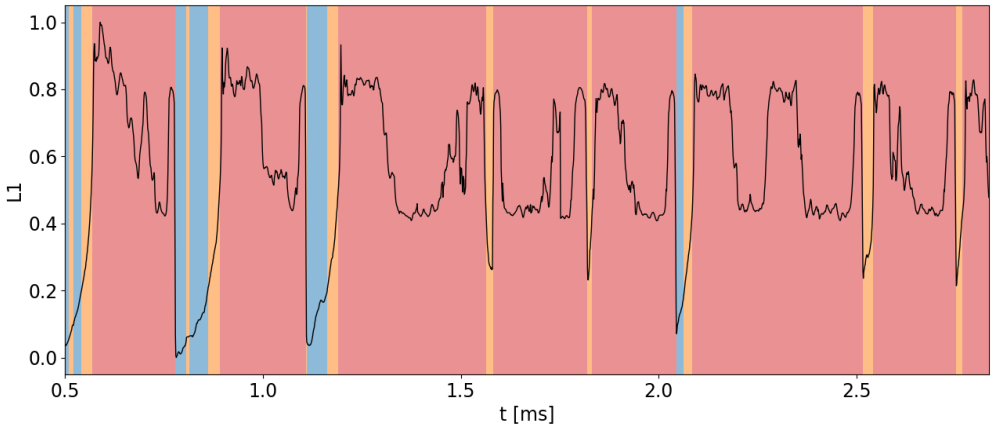


Figure 7.39: Final Time Series for S4

Figure 7.39 shows the first latent coordinate for S4. The segmentation remains quite coherent, where each extreme event is preceded by a single, precursor interval, and no missed precursors are evident. Precursor onsets align with the steep upward gradients of L1, and the return to normal typically follows the subsequent rapid decay. However, the maximum precursor time (corresponding to the second flashback as was for Figure 7.34) is only 22 μs, as opposed to 42 μs. Furthermore, two clear false positives are present before the first two flashbacks, including no normal phases for the ares preceeding flashbacks 4, 5, 7 and 8. Naturally, this is not an error of the clustering algorithm, rather a feature of the S4 location. The reason why S4 was not chosen as the primary location of investigation was due to its small relaxation time meaning a much narrower width between flashbacks as compared to S3. This makes it much harder for the clustering algorithm to understand when or if the normal event actually take place between later flashbacks. Therefore, while the final time series in this location is not perfect, the location iteself inhibits better accuracy.

7.4. Feature Feeding

Although the robustness analysis has been completed, an additional investigation is carried out. This study follows Floris [42] closely, with the main differences being the sampling at a practical location and the use of an autoencoder to extract features from the simulation variables. Specifically, data are taken from the wall, passed through the autoencoder to obtain latent variables, and then supplied to the clustering algorithm.

To assess whether the autoencoder is truly necessary, an alternative approach is considered in which the raw features from S3 are provided directly to the clustering algorithm. For this purpose, the best features identified by Floris [42] are used, with the same extreme threshold of $1300\,\mathrm{K}$ for the temperature. The selected features are temperature, pressure, x-velocity, density, and the natural logarithm of Y_{OH} (which was shown to be more accurate). In total, five features are input directly into the algorithm. The results are presented below, with plots of pressure and temperature shown for comparison.

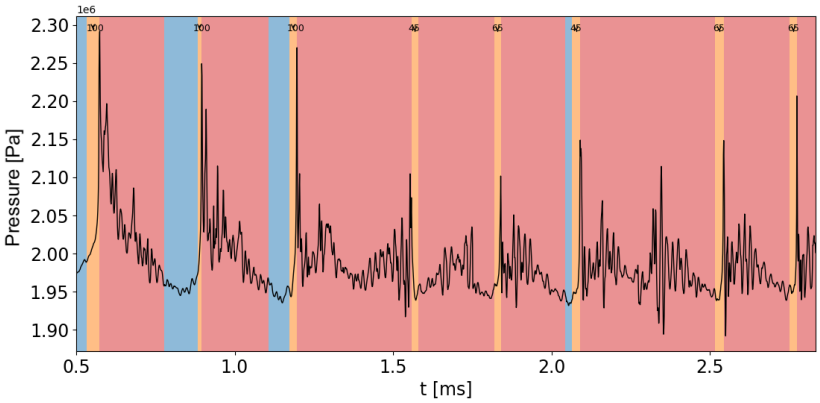


Figure 7.40: Clustering Results on the Time Series of Pressure

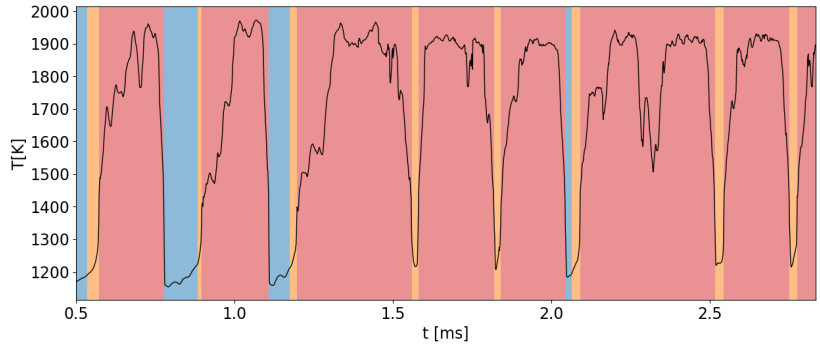


Figure 7.41: Clustering Results on the Time Series of Temperature

Figure 7.41 and Figure 7.40 show the result of the partitioning. To make it clear, temperature was used as an extreme-value threshold to generate the clusters. These clusters were then mapped onto the pressure time series, allowing for a direct comparison of turbulent features. The temperature plot shows strong agreement with the data, with almost no false positives due to the relatively low noise in the temperature signal. Every extreme event is preceded by a precursor cluster, with the importance that the 4th, 5th, 7th, and 8th flashbacks, are preceded only by a precursor cluster and not by a normal cluster.

From these observations, it can be concluded that the autoencoder and the temperature-based approach yield comparable accuracy. However, there is a significant difference in precursor duration. In the temperature plot, the average precursor time is only $20, \mu s$, with a maximum of $27, \mu s$, compared to $42, \mu s$ when using the autoencoder. While the temperature signal captures the extreme events with high accuracy, the autoencoder inherently produces longer precursors because it draws from a broader set of features.

When comparing the temperature-only case in Figure 7.41 to the autoencoder in Figure 7.35, it becomes evident that precursors in the temperature approach occur only when the temperature begins to rise. In addition, while the latent variable selected for the extreme-value threshold closely resembles the temperature, its behavior during the relaxation phases is noticeably different. Specifically, the temperature signal remains largely stagnant during relaxation and only begins to rise once a precursor event is initiated. In contrast, the latent variable produced by the autoencoder does not remain flat in these phases; instead, it exhibits a slight upward gradient. This subtle increase allows the autoencoder

to identify precursor activity earlier. Thus, it is not that the autoencoder interprets stagnant regions as precursors, but rather that its latent representation avoids complete stagnation, enabling the detection of precursors that the raw temperature signal alone would miss. This suggests that although temperature alone is a strong indicator, the autoencoder provides an earlier warning of extreme events by leveraging more comprehensive information from the system.

It should also be noted that, when comparing the flashbacks between the two approaches, the autoencoder identifies a greater proportion of flashbacks that are preceded by both a normal and a precursor cluster. In contrast, the temperature-based plot shows a larger number of flashbacks that are preceded only by a precursor cluster and lack a corresponding normal cluster. This distinction highlights the broader sensitivity of the autoencoder, which is able to capture a more complete sequence of cluster transitions leading into an extreme event. This investigation, therefore, makes a strong case for the use of an autoencoder to capture more information. Finally, the flashback will attempt to be supressed in the next section.

7.5. Flashback Suppression

In this final section, the flashback suppression attempt will be shown. As a reminder, the parameters used are in Table 5.4. The second flashback in Figure 7.34 had a precursor time of 42 μs, therefore the sprays will be turned on 42 μs before the time corresponding to the jump in temperature in Figure 7.35. The outcome is shown in Figure 7.42. To help understand the timescales, the subplots' times are shown in Table 7.4

Table 7.4: Subplot Timestamps and Deltas

Item	t **[s]**	t **[ms]**	Δt **from spray start [μs]**
(a)	0.00259705	2.59705	-195.95
Spray start	0.00279300	2.79300	0.00
(b)	0.00282706	2.82706	34.06
(c)	0.00288305	2.88305	90.05
(d)	0.00293008	2.93008	137.08

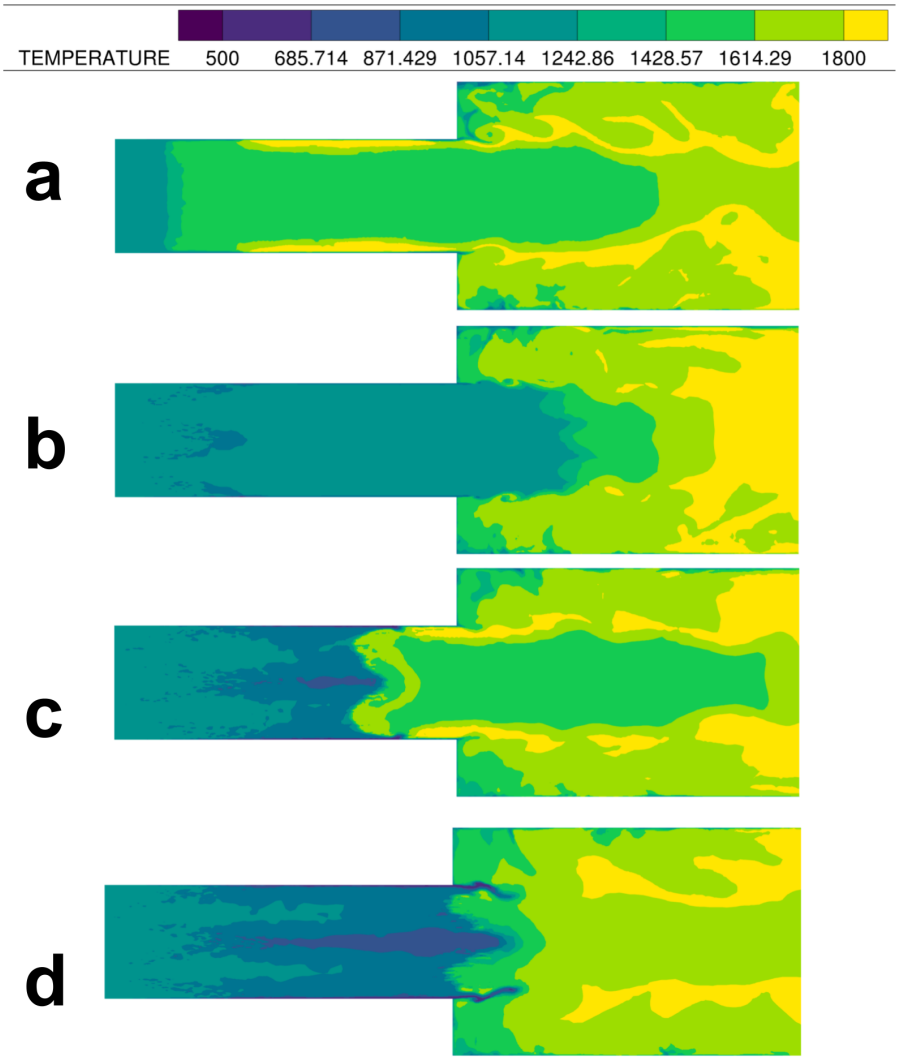


Figure 7.42: Flashback Suppression Attempt

Figure 7.42 presents temperature fields at four instants relative to the spray-start time. Panel (a) documents the end of the preceding cycle: the flashback has penetrated almost to the inlet, after which the system relaxes. By (b) the sprays have already begun ($+34$ μs after t_s) and, although little or no flashback is apparent to the eye, the precursor detector has already triggered, providing an early warning before a visible intrusion develops. In (c) ($+70$ μs) the system is in the extreme phase: the spray has propagated slightly upstream in the mixing duct and has met the flashback front; a brief upstream excursion of the flame is arrested as the spray pushes it downstream. Finally, (d) ($+117$ μs) shows the front driven fully back into the combustion chamber, illustrating effective flashback suppression on $\mathcal{O}(10^2\ \mu\text{s})$ time scales. The measurements here are taken at S3, near the spray–flashback interaction in panel c. Choosing a sampling point further upstream, e.g., at the inlet, reduces the attainable lead time because the “extreme” state is defined by the flashback reaching that location; in such a configuration the front necessarily arrives sooner at the sensor, leaving a smaller precursor window.

Overall, the empirical approach produced a satisfactory spray design capable of effectively suppressing flashback. Nevertheless, the design is still preliminary and leaves considerable room for improvement. In this study, the water velocity approached the speed of sound in the mixture, a challenging condition to achieve in practice. Since this work remains theoretical, future improvements could include extending

the precursor time to allow injection at lower velocities. Injecting the water more slowly would not only be more practical, but also enable tapering of the injection rate so that the spray does not contribute to the piston effect or promote the formation of ignition kernels.

8

Conclusion and Recommendations

8.1. Conclusion

This thesis has investigated the prediction and mitigation of flashback in hydrogen combustion, with particular emphasis on data-driven detection methods that leverage practical monitoring locations. Using a simplified model of Ansaldo Energia's GT36 reheat combustor at 20 bar, Large Eddy Simulation (LES) was performed under lean, premixed conditions. The simulations employed the Thickened Flame Model (TFM) for combustion closure, the SAGE detailed chemistry solver with a hydrogen–air mechanism to capture radical pathways, and Navier–Stokes Characteristic Boundary Conditions (NSCBC) to ensure stable wave propagation at the inlets and outlets. Together, these modeling choices provided a high-fidelity description of the coupled fluid–chemistry dynamics while keeping the computational cost reasonable. The resulting LES revealed repeated autoignition-driven flashback events in the mixing duct, triggered by the convergence of high-amplitude pressure waves that reflect off combustor walls and induce compressive heating. These waves accelerate chemical kinetics in the boundary layers, leading to early autoignition and flame propagation. Once expelled back into the combustor, the process repeats, establishing a self-sustained cycle of autoigntion, propagation, and expulsion. This recurring behaviour created a challenging but representative dataset, heavy in nonlinear dynamics, that could be used to test and develop precursor detection strategies.

The proposed framework integrates dimensionality reduction through deep autoencoders with state identification via modularity-based clustering. Fourteen thermodynamic, velocity, and species mass fraction variables were extracted from locations on the combustor wall, rather than at the flame front. This choice reflects a deliberate step towards practical applicability, since wall-mounted probes for pressure or temperature are feasible in industrial combustors, whereas intrusive in-flame measurements are not. A preliminary analysis of the extracted features showed that, compared to flame-front signals, the wall data are substantially noisier and exhibit stronger high-frequency fluctuations. Nevertheless, the dominant cyclic behaviour associated with flashback and recovery phases remains clearly imprinted in variables such as temperature T, density ρ, and key species including Y_{OH} and $Y_{\mathrm{HO_2}}$. Pressure P and velocity components (u, v, w) displayed higher levels of noise but still preserved phase-locked modulation tied to flashback events. This combination of noise and structure makes wall features more challenging to model but also more representative of realistic monitoring conditions, thereby justifying their use as the primary data source for precursor detection.

Autoencoders with two, three, and four latent variables were trained to compress the high-dimensional,

correlated wall-based time series into compact representations. Each network consisted of a feed-forward encoder–decoder architecture with strictly decreasing hidden layer widths, linear activations in the encoder, and a sigmoid output layer to account for the min–max scaling of the input data. Training was performed for 50 epochs with mean squared error (MSE) as the loss function, optimised via Adam, and with both L1 and L2 regularisation terms applied to discourage overfitting and enforce sparsity where beneficial. The two-latent bottleneck was found to be under-parameterised, encoding only the dominant cyclic envelope while systematically discarding high-frequency excursions and sharp transients. A three-latent model yielded the largest improvement: the added dimension isolated transition sharpness and mid-frequency structure, while regularisation values near zero indicated that all latent coordinates were actively utilised. A fourth latent dimension produced incremental refinements, particularly in noisy velocity and radical channels, but did so under stronger latent activity penalties, indicating diminishing returns. Taken together, these results established the three-latent model as the optimal trade-off between expressiveness and interpretability, providing a stable, low-dimensional, and physically meaningful representation of system dynamics.

Building on this reduced representation, a modularity-based clustering algorithm was applied to detect precursors of flashback. The latent trajectories were tessellated into hypercubes, mapped into a graph, and clustered by maximizing modularity to identify distinct dynamical states. To improve robustness, an extreme-value filter was applied: points exceeding a threshold of 0.25 in the latent coordinate most strongly correlated with temperature were classified instantly as extreme. This latent was selected because temperature exhibits the clearest cyclic behaviour and thus provides a stable reference for distinguishing genuine precursor dynamics from background variability. The approach proved highly effective: precursor states were consistently identified with a maximum lead time of approximately 42 μs, sufficient for active control. Importantly, the method achieved this with only one small false positive at the fast flashback regime, demonstrating its reliability in discriminating genuine precursors from background fluctuations. Therefore, clustering enabled the successful prediction and subsequent suppression of an oncoming flashback, demonstrating the framework's practical applicability for real-time instability mitigation.

The robustness of the framework was evaluated through several tests. Varying latent dimensionality for clustering showed that two latents caused a low false positive rate, but a correspondingly low precursor time, while four latents caused a fragmented clustering scheme; three latents remained the optimal balance. Threshold sensitivity analysis confirmed that precursor onset was unaffected by changing the extreme-value cutoff (0.22–0.28), proving that the chosen 0.25 had negligible impact on the early-warning index. Tests on unseen data segments demonstrated reliable online prediction, with slightly shorter precursor durations (23 µs) but consistent detection. Using an unseen monitoring location (S4) confirmed that the autoencoder generalized to new signals, though warning times shortened (22 µs) and false positives rose, reflecting noisier dynamics. A feature-feeding comparison showed that raw variables without an autoencoder reproduced precursors but with shorter precursor times (20 µs), while autoencoder latents achieved earlier and more stable predictions (42 µs).

Finally, the precursor window was used for active control. Triggered sprays arrested and then expelled an advancing flashback within 100 µs, proving that the method not only predicts but also enables suppression.

Overall, this thesis shows that by combining wall-based sensing, deep autoencoder representations, and modularity-based clustering, it is possible to reliably detect flashback precursors in hydrogen combustion with actionable lead times. The demonstration of precursor-based suppression illustrates that data-driven approaches can move beyond offline diagnostics to active intervention, offering a step forward in ensuring safe and stable operation of next-generation hydrogen gas turbines.

8.2. Recommendations

This study demonstrated that precursor detection and suppression of flashback in hydrogen combustion can be achieved using wall-based sensor data, autoencoder-based dimensionality reduction, and modularity-driven clustering. Building on these findings, several directions for future work can be identified:

- **Extreme value definition.** In this work, the extreme cluster was defined by a fixed threshold (0.25) on the latent variable most strongly correlated with temperature. While sensitivity tests showed that this choice had little influence on the earliest precursor index, a more general definition could be sought. For example, gradient-based metrics across all latent variables could be used to identify sharp excursions without predefining a value. Although challenging due to the nonlinear behaviour of the latents, such an approach could exploit the full latent space rather than a single coordinate.
- **Inlet fluctuations.** The LES setup imposed steady inlet conditions. Introducing synthetic turbulence through digital filter methods or similar techniques would provide a stronger test of robustness under fluctuating inflow, which is expected in real combustors. This is particularly relevant since the wall signals already exhibit significant noise, and additional unsteadiness would probe the algorithm's limits.
- **Multi-location encoding.** All analyses here were based on a single wall location. A potential extension would be to incorporate multiple monitoring points. One possibility is a hierarchical autoencoder: each location is compressed individually, followed by a second-stage encoder that combines them into a unified latent representation. This could yield a more global view of system dynamics, though at the cost of added complexity.
- **Flashback suppression.** The spray suppression case was exploratory and not systematically optimised. In practice, the design of injectors (diameter, cone angle, velocity, droplet size distribution) requires detailed investigation. Moreover, the successful case in this work relied on near-sonic injection velocities, which are unlikely to be feasible in realistic combustors. Longer precursor times or alternative actuation methods would relax these requirements and should be studied further.
- **Flashback sequencing.** The dataset used here was generated under conditions chosen to ensure multiple flashbacks. As a result, once initiated, flashbacks occurred quasi-periodically, which may have simplified detection. In reality, the first flashback in a combustor is the most critical, as subsequent ones are influenced by prior events. Future work should therefore focus on detection and suppression at the very first occurrence, which represents the truest analogue to operational conditions.

Through these avenues, the proposed approach can move from proof-of-concept to a deployable early-warning and control system, supporting safer and more reliable hydrogen-based combustion technologies.

References

[1] S. S. Abdurakipov et al. “Combustion Regime Monitoring by Flame Imaging and Machine Learning”. In: *Optoelectronics, Instrumentation and Data Processing* (), pp. 513–519. DOI: 10.3103/S875669901805014X.

[2] B. Abramzon and W. Sirigano. “Droplet vaporization model for spray combustion calculations”. In: *26th Aerospace Sciences Meeting* (). DOI: 10.2514/6.1988-636.

[3] Acciona. *What Are The Colours Of Hydrogen And What Do They Mean?* Acciona. URL: https://www.acciona.com.au/updates/stories/what-are-the-colours-of-hydrogen-and-what-do-they-mean/.

[4] Konduri Aditya et al. “Direct numerical simulation of flame stabilization assisted by autoignition in a reheat gas turbine combustor”. In: *Proceedings of the Combustion Institute* (), pp. 2635–2642. DOI: 10.1016/j.proci.2018.06.084.

[5] R. Adnan, H. H. Masjuki, and T. M. I. Mahlia. “Performance and emission analysis of hydrogen fueled compression ignition engine with variable water injection timing”. In: *Energy* (), pp. 416–426. DOI: 10.1016/j.energy.2012.03.073.

[6] R. Amaduzzi et al. “Effect of parametric uncertainty in numerical simulations of a hydrogen-fueled flameless combustion furnace using dimensionality reduction and non-linear regression”. In: *Proceedings of the Combustion Institute* (), p. 105551. DOI: 10.1016/j.proci.2024.105551.

[7] E. Amani, M. R. Akbari, and S. Shahpouri. “Multi-objective CFD optimizations of water spray injection in gas-turbine combustors”. In: *Fuel* (), pp. 267–278. DOI: 10.1016/j.fuel.2018.04.093.

[8] A. A. Amsden, P. J. O’Rourke, and T. D. Butler. *KIVA-II: A computer program for chemically reactive flows with sprays*. DOI: 10.2172/6228444. URL: https://www.osti.gov/biblio/6228444.

[9] Ansaldo. *GT36 sequential combustion technology achieves 100% hydrogen*. URL: https://www.ansaldoenergia.com/about-us/media-center/power-generation-news-insights/detail-news/gt36-sequential-combustion-technology-achieves-100-hydrogen.

[10] Anna Asch et al. “Model-assisted deep learning of rare extreme events from partial observations”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* (), p. 043112. DOI: 10.1063/5.0077646.

[11] N. Ashgriz and J. Y. Poo. “Collision Dynamics of Two Liquid Drops”. In: *Physics of Fluids A: Fluid Dynamics* (), pp. 1446–1446. DOI: 10.1063/1.4738835.

[12] Xiaojing Bai et al. “Multi-mode combustion process monitoring on a pulverised fuel combustion test facility based on flame imaging and random weight network techniques”. In: *Fuel* (), pp. 656–664. DOI: 10.1016/j.fuel.2017.03.091.

[13] Jadeed Beita et al. “Thermoacoustic Instability Considerations for High Hydrogen Combustion in Lean Premixed Gas Turbine Combustors: A Review”. In: *Hydrogen* (), pp. 33–57. DOI: 10.3390/hydrogen2010003.

[14] Ivan Belcic. *What is Supervised Learning?* IBM. URL: https://www.ibm.com/think/topics/supervised-learning.

[15] Ali Cemal Benim and Khawar J. Syed. “An Overview of Flashback Mechanisms”. In: *Flashback Mechanisms in Lean Premixed Gas Turbine Combustion* (), pp. 25–26. DOI: 10.1016/B978-0-12-800755-6.00004-0.

[16] Ali Cemal Benim and Khawar J. Syed. “Combustion-Induced Vortex Breakdown–Driven Flashback”. In: *Flashback Mechanisms in Lean Premixed Gas Turbine Combustion* (), pp. 73–102. DOI: 10.1016/B978-0-12-800755-6.00009-X.

[17] Ali Cemal Benim and Khawar J. Syed. “Flashback by Autoignition”. In: *Flashback Mechanisms in Lean Premixed Gas Turbine Combustion* (), pp. 27–39. DOI: 10.1016/B978-0-12-800755-6.00005-2.

[18] Lukas Berger et al. “Flame fingers and interactions of hydrodynamic and thermodiffusive instabilities in laminar lean hydrogen flames”. In: *Proceedings of the Combustion Institute* (), pp. 1525–1534. DOI: 10.1016/j.proci.2022.07.010.

[19] Blackridge. *Global Top 15 Gas Turbine Manufacturers 2025*. Blackridge Research. URL: https://www.blackridgeresearch.com/blog/list-of-global-top-gas-turbine-manufacturers-makers-companies-installers-suppliers-in-the-world/.

[20] Mirko Bothien, John Wood, and Gerhard Fruechtel. “Toward Decarbonized Power Generation With Gas Turbines by Using Sequential Combustion for Burning Hydrogen”. In: *Journal of Engineering for Gas Turbines and Power* (), pp. 121013–1. DOI: 10.1115/1.4045256.

[21] Thomas M. Bury et al. “Deep learning for early warning signals of tipping points”. In: *Proceedings of the National Academy of Sciences* (), e2106140118. DOI: 10.1073/pnas.2106140118.

[22] M. S. Butler et al. “Limits for hydrogen leaks that can support stable flames”. In: *International Journal of Hydrogen Energy* (), pp. 5174–5182. DOI: 10.1016/j.ijhydene.2009.04.012.

[23] Lucas de Carvalho Pagliosa. “Exploring chaotic time series and phase spaces: from dynamical systems to visual analytics”. PhD thesis. University of Groningen. DOI: 10.33612/diss.117450127.

[24] K. Champion, S. L. Brunton, and J. N. Kutz. “Data-driven discovery of coordinates and governing equations”. In: *Proceedings of the National Academy of Sciences* (), pp. 22171–22176. DOI: 10.1073/pnas.1904220116.

[25] Fabrice Charlette, Charles Meneveau, and Denis Veynante. “A power-law flame wrinkling model for LES of premixed turbulent combustion Part I: non-dynamic formulation and initial tests”. In: *Combustion and Flame* (), pp. 159–180. DOI: 10.1016/S0010-2180(02)00400-5.

[26] Junjie Chen and Tengfei Li. “Combustion Characteristics of Methane-Air Mixtures in Millimeter-Scale Systems With a Cavity Structure: An Experimental and Numerical Study”. In: *Frontiers in Energy Research* (). DOI: 10.3389/fenrg.2022.807902.

[27] Andrea Ciani et al. “Superior fuel and operational flexibility of sequential combustion in Ansaldo Energia gas turbines”. In: *Journal of the Global Power and Propulsion Society* (), pp. 630–638. DOI: 10.33737/jgpps/110717.

[28] O. Colin et al. “A thickened flame model for large eddy simulations of turbulent premixed combustion”. In: *Physics of Fluids* (), pp. 1843–1863. DOI: 10.1063/1.870436.

[29] The European Commission. “Commission Implementing Decision (EU) 2021/2326 of 30 November 2021 establishing best available techniques (BAT) conclusions, under Directive 2010/75/EU of the European Parliament and of the Council, for large combustion plants”. In: *Official Journal of the European Union* (). URL: https://eur-lex.europa.eu/eli/dec_impl/2021/2326/oj.

[30] Riccardo Concetti et al. “Effects of liquid water addition on turbulent premixed hydrogen/air combustion”. In: *Fuel* (), p. 132314. DOI: 10.1016/j.fuel.2024.132314.

[31] Inc. Convergent Science. *CONVERGE CFD v3.1 Manual*. URL: https://convergecfd.com.

[32] Bidhan Dam, Norman Love, and Ahsan Choudhuri. “Flashback propensity of syngas fuels”. In: *Fuel* (), pp. 618–625. DOI: 10.1016/j.fuel.2010.10.021.

[33] Siyu Ding et al. “Reduced-order modeling via convolutional autoencoder for emulating combustion of hydrogen/methane fuel blends”. In: *Combustion and Flame* (), p. 113981. DOI: 10.1016/j.combustflame.2025.113981.

[34] Tarek Echekki and Jacqueline H. Chen. “Direct numerical simulation of autoignition in non-homogeneous hydrogen-air mixtures”. In: *Combustion and Flame* (), pp. 169–191. DOI: 10.1016/S0010-2180(03)00088-9.

[35] Ember. *Analysis of key power sector emitters in 2023*. Ember. URL: https://ember-energy.org/latest-insights/global-electricity-review-2024/major-countries-and-regions.

[36] Ansaldo Energia. *GT36 - The superior value*. Ansaldo Energia. URL: https://www.ansaldoenergia.com/offering/equipment/turbomachinery/gt36.

[37] Ansaldo Energia. *Hydrogen Technology*. URL: https://www.ansaldoenergia.com/fileadmin/Brochure/Review_2023/AnsaldoEnergia-HydrogenTechnology-20230927.pdf.

[38] US Department of Energy. *Hydrogen Storage*. US Department of Energy. URL: https://www.energy.gov/eere/fuelcells/hydrogen-storage.

[39] Ikeuchi Europe. *What is an air atomizing spray nozzle?* Ikeuchi Europe. URL: https://www.ikeuchi.eu/news/what-is-an-air-atomizing-spray-nozzles/.

[40] A. Farokhipour, E. Hamidpour, and E. Amani. “A numerical study of NOx reduction by water spray injection in gas turbine combustion chambers”. In: *Fuel* (), pp. 173–186. DOI: 10.1016/j.fuel.2017.10.033.

[41] A. Fichera, C. Losenno, and A. Pagano. “Clustering of chaotic dynamics of a lean gas-turbine combustor”. In: *Applied Energy* (), pp. 101–117. DOI: 10.1016/S0306-2619(00)00067-2.

[42] Mihnea Floris. *Flashback Detection and Suppression in Reheat Hydrogen Combustor*. URL: https://repository.tudelft.nl/record/uuid:7f0ac3d0-9a1a-492c-8e9f-1cb2be06774d.

[43] Mihnea Floris et al. “Data-driven identification of precursors of flashback in a lean hydrogen reheat combustor”. In: *Proceedings of the Combustion Institute* (), p. 105524. DOI: 10.1016/j.proci.2024.105524.

[44] J. Fritz, M. Kröner, and T. Sattelmayer. “Flashback in a swirl burner with cylindrical premixing zone”. In: *Journal of Engineering for Gas Turbines and Power* (), pp. 276–283. DOI: 10.1115/1.1473155.

[45] GeeksforGeeks. *Transition Probability Matrix*. URL: https://www.geeksforgeeks.org/engineering-mathematics/transition-probability-matrix/.

[46] Urszula Golyska and Nguyen Anh Khoa Doan. “Clustering-Based Identification of Precursors of Extreme Events in Chaotic Systems”. In: *Lecture Notes in Computer Science* (), pp. 313–327. DOI: 10.1007/978-3-031-36027-5_23.

[47] Ana González-Espinosa et al. “Effects of hydrogen and primary air in a commercial partially-premixed atmospheric gas burner by means of optical and supervised machine learning techniques”. In: *International Journal of Hydrogen Energy* (), pp. 31130–31150. DOI: 10.1016/j.ijhydene.2020.08.045.

[48] Grammarly Blog Team. *What Is an Autoencoder in Deep Learning?* URL: https://www.grammarly.com/blog/ai/what-is-autoencoder/.

[49] Hartmut Grassl and Dietrich Brockhagen. “Climate forcing of aviation emissions in high altitudes and comparison of metrics: An update according to the Fourth Assessment Report, IPCC 2007”. In: *Unpublished manuscript (Max Planck Institute for Meteorology / atmosfair)* (2007).

[50] Andrea Gruber et al. “A Numerical Investigation of Reheat Hydrogen Combustion in a Simplified Geometrical Configuration From Atmospheric Pressure to Full Load Conditions”. In: *ASME Turbo Expo* (), V03BT04A045. DOI: 10.1115/GT2022-83218.

[51] Andrea Gruber et al. “Direct Numerical Simulation of hydrogen combustion at auto-ignitive conditions: Ignition, stability and turbulent reaction-front velocity”. In: *Combustion and Flame* (), p. 111385. DOI: 10.1016/j.combustflame.2021.02.031.

[52] Feng Han, Shuying Yang, and Shibao Song. “Local Volterra multivariable chaotic time series multi-step prediction based on phase points clustering”. In: *Journal of Vibroengineering* (), pp. 2486–2503. DOI: 10.21595/jve.2018.19142.

[53] Zhezhe Han et al. “Combustion stability monitoring through flame imaging and stacked sparse autoencoder based deep neural network”. In: *Applied Energy* (), p. 114159. DOI: 10.1016/j.apenergy.2019.114159.

[54] Zhezhe Han et al. “Prediction of combustion state through a semi-supervised learning model and flame imaging”. In: *Fuel* (), p. 119745. DOI: 10.1016/j.fuel.2020.119745.

[55] A. Hanuschkin et al. “Investigation of cycle-to-cycle variations in a spark-ignition engine based on a machine learning analysis of the early flame kernel”. In: *Proceedings of the Combustion Institute* (), pp. 5751–5759. DOI: 10.1016/j.proci.2020.05.030.

[56] Josef Hasslberger et al. “Physical effects of water droplets interacting with turbulent premixed flames: A Direct Numerical Simulation analysis”. In: *Combustion and Flame* (), p. 111404. DOI: 10.1016/j.combustflame.2021.111404.

[57] John B. Heywood. *Internal Combustion Engine Fundamentals*, p. 915.

[58] Shuhai Hou and David P. Schmidt. “Interaction Mechanisms between Closely Spaced Sprays”. In: *SAE Technical Paper* (), pp. 2008-01–0946. DOI: 10.4271/2008-01-0946.

[59] IBM. *What is Machine Learning?* IBM. URL: https://www.ibm.com/think/topics/machine-learning.

[60] IBM. *What is Unsupervised Learning?* IBM. URL: https://www.ibm.com/think/topics/unsupervised-learning.

[61] Hassaan Idrees. *Autoencoders vs. PCA: Dimensionality Reduction for Complex Data*. Medium. URL: https://medium.com/@hassaanidrees7/autoencoders-vs-pca-dimensionality-reduction-for-complex-data-e07d4612b711.

[62] IEA. *Aviation*. International Energy Agency. URL: https://www.iea.org/energy-system/transport/aviation.

[63] Kazuki Iemura et al. “Analysis of spatial-temporal dynamics of cool flame oscillation phenomenon occurred around a fuel droplet array by using variational auto-encoder”. In: *Proceedings of the Combustion Institute* (), pp. 2523–2532. DOI: 10.1016/j.proci.2022.09.047.

[64] R. I. Issa. “Solution of the implicitly discretised fluid flow equations by operator-splitting”. In: *Journal of Computational Physics* (), pp. 40–65. DOI: 10.1016/0021-9991(86)90099-9.

[65] Herbert Jaeger and Harald Haas. “Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication”. In: *Science* (), pp. 78–80. DOI: 10.1126/science.1091277.

[66] Chenxi Ji et al. “Development of novel combustion risk index for flammable liquids based on unsupervised clustering algorithms”. In: *Journal of Loss Prevention in the Process Industries* (), p. 104422. DOI: 10.1016/j.jlp.2021.104422.

[67] Jones et al. “National Contributions to Climate Change Due to Historical Emissions of Carbon Dioxide, Methane and Nitrous Oxide”. In: *Zenodo* (). DOI: 10.5281/zenodo.14054503.

[68] Anirudh Jonnalagadda et al. “A co-kurtosis based dimensionality reduction method for combustion datasets”. In: *Combustion and Flame* (), p. 112635. DOI: 10.1016/j.combustflame.2023.112635.

[69] Zhe Kang et al. “Effect of direct water injection timing on cycle performance and emissions characteristics within a CI-ICRC engine”. In: *Alexandria Engineering Journal* (), pp. 135–145. DOI: 10.1016/j.aej.2023.03.087.

[70] Holger Kantz and Thomas Schreiber. “Instability: Lyapunov Exponents”. In: *Nonlinear Time Series Analysis* (), pp. 65–74.

[71] D. Karaoulanis. *Chaotic time series & forecasting*. University of Athens. URL: https://prognostikon.cce.uoa.gr/dkaraoulanis/chaotic-time-series-forecasting/.

[72] Lyudmyla Kirichenko, Oksana Pichugina, and Hlib Zinchenko. “Clustering Time Series of Complex Dynamics by Features”. In: *International Conference on Information Technology and Interactions* (). DOI: 10.5281/zenodo.248507660.

[73] Boris Kruljevic et al. “LES/Thickened Flame Model of Reheat Hydrogen Combustion With Water/Steam Injection”. In: *ASME Turbo Expo: Turbomachinery Technical Conference and Exposition* (), V03BT04A042. DOI: 10.1115/GT2023-103466.

[74] Thierry Lecomte et al. “Best Available Techniques (BAT) Reference Document for Large Combustion Plants”. In: *JRC Science for Policy Report* ().

[75] Arthur H. Lefebvre and Vincent G. McDonell. *Atomization and Sprays*, p. 300. DOI: 10.1201/9781315120911.

[76] J. Legier, Thierry Poinsot, and D. Veynante. “Dynamically thickened flame LES model for premixed and non-premixed turbulent combustion”. In: *Proceedings of the Summer Program* ().

[77] E. A. Leicht and M. E. J. Newman. “Community Structure in Directed Networks”. In: *Physical Review Letters* (), p. 118703. DOI: 10.1103/PhysRevLett.100.118703.

[78] Juan Li et al. “An updated comprehensive kinetic model of hydrogen combustion”. In: *International Journal of Chemical Kinetics* (), pp. 566–575. DOI: 10.1002/kin.20026.

[79] Alex B. Liu, Daniel Mather, and Rolf D. Reitz. “Modeling the Effects of Drop Drag and Breakup on Fuel Sprays”. In: *SAE Technical Paper* (), p. 930072. DOI: 10.4271/930072.

[80] Jie Liu et al. “On explosion-limit regime diagram of H2 and C1 to C3 alkanes with unified pivot state”. In: *Combustion and Flame* (), p. 112705. DOI: 10.1016/j.combustflame.2023.112705.

[81] Mantas Lukoševičius and Herbert Jaeger. “Reservoir computing approaches to recurrent neural network training”. In: *Computer Science Review* (), pp. 127–149. DOI: 10.1016/j.cosrev.2009.03.005.

[82] Lindsay Maizland and Clara Fong. *Global Climate Agreements: Successes and Failures*. Council on Foreign Relations. URL: https://www.cfr.org/backgrounder/paris-global-climate-change-agreements?.

[83] Mohammad Rafi Malik et al. “Dimensionality reduction and unsupervised classification for high-fidelity reacting flow simulations”. In: *Proceedings of the Combustion Institute* (), pp. 5155–5163. DOI: 10.1016/j.proci.2022.06.017.

[84] A. Manuel et al. “A Study of Diesel Cold Starting using both Cycle Analysis and Multidimensional Calculations”. In: *SAE Technical Paper* (), p. 910180. DOI: 10.4271/910180.

[85] Norbert Marwan et al. “Recurrence plots for the analysis of complex systems”. In: *Physics Reports* (), pp. 237–329. DOI: 10.1016/j.physrep.2006.11.001.

[86] Michael McCartney, Thomas Indlekofer, and Wolfgang Polifke. “Online Detection of Combustion Instabilities Using Supervised Machine Learning”. In: *ASME Turbo Expo: Turbomachinery Technical Conference and Exposition* (), V04AT04A045. DOI: 10.1115/GT2020-14834.

[87] S. Menon, P.-K. Yeung, and W.-W. Kim. “Effect of subgrid models on the computed interscale energy transfer in isotropic turbulence”. In: *Computers & Fluids* (), pp. 165–180. DOI: 10.1016/0045-7930(95)00036-4.

[88] Mitsubishi. *M501J Series*. Mitsubishi Power. URL: https://power.mhi.com/regions/amer/products/gas-turbines/m501j.

[89] J. D. Naber and Rolf D. Reitz. “Modeling Engine Spray/Wall Impingement”. In: *SAE Technical Paper* (), p. 880107. DOI: 10.4271/880107.

[90] M. E. J. Newman. “Modularity and community structure in networks”. In: *Proceedings of the National Academy of Sciences* (), pp. 8577–8582. DOI: 10.1073/pnas.0601602103.

[91] European Parliament. *Greenhouse gas emissions by country and sector*. European Parliament. URL: https://www.europarl.europa.eu/pdfs/news/expert/2018/3/story/20180301STO98928/20180301STO98928_en.pdf.

[92] J. Pathak et al. “Model-free prediction of large spatiotemporally chaotic systems from data”. In: *Proceedings of the National Academy of Sciences* (), pp. 1051–1056. DOI: 10.1073/pnas.1717005115.

[93] T. Poinsot et al. *Simulation tools for 3D reacting flows*. URL: https://cefrc.princeton.edu/sites/g/files/toruqf1071/files/Files/2013%20Lecture%20Notes/Poinsot/5-Codes-DNS-LES-RANS.pptx.pdf.

[94] T. J. Poinsot and S. K. Lelef. “Boundary conditions for direct simulations of compressible viscous flows”. In: *Journal of Computational Physics* (), pp. 104–129. DOI: 10.1016/0021-9991(92)90046-2.

[95] Eric Pomraning. “Development of Large Eddy Simulation Turbulence Models”. PhD thesis. University of Wisconsin–Madison. DOI: 10.13140/2.1.2035.7929.

[96] Stephen B. Pope. “Ten questions concerning the large-eddy simulation of turbulent flows”. In: *New Journal of Physics* (), p. 35. DOI: 10.1088/1367-2630/6/1/035.

[97] Scott L. Post and John Abraham. “Modeling the outcome of drop–drop collisions in Diesel sprays”. In: *International Journal of Multiphase Flow* (), pp. 997–1019. DOI: 10.1016/S0301-9322(02)00007-1.

[98] Bhukya Ramadevi and Kishore Bingi. “Chaotic Time Series Forecasting Approaches Using Machine Learning Techniques: A Review”. In: *Symmetry* (), p. 955. DOI: 10.3390/sym14050955.

[99] Rolf Deneys Reitz. “Atomization and Other Breakup Regimes of a Liquid Jet”. PhD thesis. Princeton University. URL: https://ui.adsabs.harvard.edu/abs/1978PhDT.....69R/.

[100] UK Research and Innovation. *A brief history of climate change discoveries*. UKRI. URL: https://www.discover.ukri.org/a-brief-history-of-climate-change-discoveries.

[101] C. M. Rhie and W. L. Chow. “Numerical study of the turbulent flow past an airfoil with trailing edge separation”. In: *AIAA Journal* (), pp. 1525–1532. DOI: 10.2514/3.8284.

[102] Hannah Ritchie. *What share of global CO□ emissions come from aviation?* Our World in Data. URL: https://ourworldindata.org/global-aviation-emissions.

[103] Hannah Ritchie and Pablo Rosado. “Energy Mix”. In: *Our World in Data* ().

[104] Pablo Rouco Pousada et al. “Flashback Prevention in a Hydrogen-Fueled Reheat Combustor by Water Injection Optimized With Global Sensitivity Analysis”. In: *Journal of Engineering for Gas Turbines and Power* (), p. 061021. DOI: 10.1115/1.4066895.

[105] I. Roumeliotis and K. Mathioudakis. “Evaluation of water injection effect on compressor and engine performance and operability”. In: *Applied Energy* (), pp. 1207–1216. DOI: 10.1016/j.apenergy.2009.04.039.

[106] D. N. Rrustemi et al. “New laminar flame speed correlation for lean mixtures of hydrogen combustion with water addition under high pressure conditions”. In: *International Journal of Hydrogen Energy* (), pp. 609–617. DOI: 10.1016/j.ijhydene.2024.03.177.

[107] David H. Rudy and John C. Strikwerda. “Boundary conditions for subsonic compressible Navier-Stokes calculations”. In: *Computers & Fluids* (), pp. 327–338. DOI: 10.1016/0045-7930(81)90005-0.

[108] Eustaquio A. Ruiz et al. “Convolutional neural networks to predict the onset of oscillatory instabilities in turbulent systems”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* (), p. 093131. DOI: 10.1063/5.0056032.

[109] Antonio L. Sánchez, Eduardo Fernández-Tarrazo, and Forman A. Williams. “The chemistry involved in the third explosion limit of H2–O2 mixtures”. In: *Combustion and Flame* (), pp. 111–117. DOI: 10.1016/j.combustflame.2013.07.013.

[110] David P. Schmidt and C. J. Rutland. “A New Droplet Collision Algorithm”. In: *Journal of Computational Physics* (), pp. 62–80. DOI: 10.1006/jcph.2000.6568.

[111] Oliver Schulz et al. “A criterion to distinguish autoignition and propagation applied to a lifted methane–air jet flame”. In: *Proceedings of the Combustion Institute* (), pp. 1637–1644. DOI: 10.1016/j.proci.2016.08.022.

[112] P. K. Senecal et al. “Multi-Dimensional Modeling of Direct-Injection Diesel Spray Liquid Length and Flame Lift-off Length using CFD and Parallel Detailed Chemistry”. In: *SAE Technical Paper* (), pp. 2003-01–1043. DOI: 10.4271/2003-01-1043.

[113] Siemens. *Siemens Energy gas turbine portfolio*. URL: https://p3.aprimocdn.net/siemensenergy/f592779d-ac7d-494a-a690-b20a00ba23c3/GT-Portfolio-Brochure-2024-update-pdf_Original%20file.pdf?apr_optimization=false.

[114] SIMAP. *Air travel emissions*. SIMAP. URL: https://unhsimap.org/cmap/resources/air-travel.

[115] Kevin W. Thompson. “Time dependent boundary conditions for hyperbolic systems”. In: *Journal of Computational Physics* (), pp. 1–24. DOI: 10.1016/0021-9991(87)90041-6.

[116] Kevin W. Thompson. “Time-dependent boundary conditions for hyperbolic systems, II”. In: *Journal of Computational Physics* (), pp. 439–461. DOI: 10.1016/0021-9991(90)90152-Q.

[117] The Engineering Toolbox. *Higher Calorific Values of Common Fuels: Reference & Data*. The Engineering Toolbox. URL: https://www.engineeringtoolbox.com/fuels-higher-calorific-values-d_169.html.

[118] Stephen R. Turns. *An Introduction to Combustion: Concepts and Applications*.

[119] UNFCCC. *What is the Kyoto Protocol?* UNFCCC. URL: https://unfccc.int/kyoto_protocol.

[120] K. Varatharajan and M. Cheralathan. “Influence of fuel properties and composition on NOx emissions from biodiesel powered diesel engines: A review”. In: *Renewable and Sustainable Energy Reviews* (), pp. 3702–3710. DOI: 10.1016/j.rser.2012.03.056.

[121] GE Vernova. *Combined cycle power plant: how it works*. GE Vernova. URL: https://www.gevernova.com/gas-power/resources/education/combined-cycle-power-plants.

[122] GE Vernova. *GE Vernova’s H-Class gas turbine fleet accumulates three million operating hours*. GE Vernova. URL: https://www.gevernova.com/news/press-releases/ge-vernova-h-class-gas-turbine-fleet-accumulates-three-million-operating-hours.

[123] GE Vernova. *Hydrogen overview*. URL: https://www.gevernova.com/content/dam/gepower-new/global/en_US/downloads/gas-new-site/future-of-energy/GEA35685-GEV-Hydrogen-Overview_8.5x11_RGB_R4.pdf.

[124] L. H. J. Wachters and N. A. J. Westerling. “The heat transfer from a hot wall to impinging water drops in the spheroidal state”. In: *Chemical Engineering Science* (), pp. 1047–1056. DOI: 10.1016/0009-2509(66)85100-X.

[125] Huijiang Wang, Yang Bai, and Zhe Kang. “Numerical investigation on direct water injection characteristics under different injection and ambient conditions within oxygen/argon atmosphere”. In: *Case Studies in Thermal Engineering* (), p. 104990. DOI: 10.1016/j.csite.2024.104990.

[126] Zijun Wang et al. “Effects of water sprays on hydrogen autoignition in heated air”. In: *Process Safety and Environmental Protection* (), pp. 915–925. DOI: 10.1016/j.psep.2024.10.069.

[127] Archie West. *The CFD Development of Non-premixed Dual Fuel Combustion Diesel Engine injected by High-pressure Gas in the Cylinder Chamber*. URL: https://www.archie-west.ac.uk/projects/computational-fluid-dynamics/the-cfd-development-of-non-premixed-dual-fuel-combustion-diesel-engine-injected-by-high-pressure-gas-in-the-cylinder-chamber/.

[128] Harya Widiputra et al. “A Novel Evolving Clustering Algorithm with Polynomial Regression for Chaotic Time-Series Prediction”. In: *Neural Information Processing* (), pp. 114–121. DOI: 10.1007/978-3-642-10684-2_12.

[129] Alan Wolf et al. “Determining Lyapunov exponents from a time series”. In: *Physica D: Nonlinear Phenomena* (), pp. 285–317. DOI: 10.1016/0167-2789(85)90011-9.

[130] L. Xiao, Y. Li, and X. Zhang. “Early detection of thermoacoustic instabilities using deep learning architectures”. In: *Combustion Science and Technology* (), pp. 1207–1222. DOI: 10.1080/00102202.2020.1724444.

[131] Wanqian Xu et al. “Time-series clustering of high-speed photography based on Multi-channel Variational Autoencoder in a scramjet combustor”. In: *Applied Thermal Engineering* (), p. 126164. DOI: 10.1016/j.applthermaleng.2025.126164.

[132] Weiming Xu, Tao Yang, and Peng Zhang. *Dimensionality Reduction and Dynamical Mode Recognition of Circular Arrays of Flame Oscillators Using Deep Neural Network*. URL: https://arxiv.org/abs/2312.02462.

[133] Guotian Yang et al. “Gabor-GLCM-Based Texture Feature Extraction Using Flame Image to Predict the O2 Content and NOx”. In: *ACS Omega* (), pp. 3889–3899. DOI: 10.1021/acsomega.1c03397.

[134] A. Yoshizawa and K. Horiuti. “A Statistically-Derived Subgrid-Scale Kinetic Energy Model for the Large-Eddy Simulation of Turbulent Flows”. In: *Journal of the Physical Society of Japan* (), p. 2834. DOI: 10.1143/JPSJ.54.2834.

[135] Wenbin Yu et al. “Integrated analysis of CFD simulation data with K-means clustering algorithm for soot formation under varied combustion conditions”. In: *Applied Thermal Engineering* (), pp. 299–305. DOI: 10.1016/j.applthermaleng.2019.03.011.

[136] Zhiya Zuo. *Python implementation of Newman’s spectral methods to maximize modularity*. URL: https://github.com/zhiyzuo/python-modularity-maximization.

A

Numerical Solver

For Converge, certain numerical schemes are required. The governing equations are discretized using the finite volume method (FVM). Pressure–velocity coupling is handled with the Pressure-Implicit with Splitting of Operators (PISO) algorithm [64], with Rhie–Chow interpolation [101] to avoid pressure–velocity decoupling. At each iteration, the resulting linear systems are solved using Successive Over-Relaxation (SOR).

A.1. Finite-Volume Discretization

In the FVM framework, the domain is partitioned into control volumes whose cell-center values are advanced using face fluxes and source terms. For a transported scalar ϕ, starting from the 1D transport equation and applying the divergence theorem,

$$\frac{\partial \phi}{\partial t} + \frac{\partial (u\phi)}{\partial x} = 0 \;\; \Rightarrow \;\; \frac{\partial \phi}{\partial t} + \frac{1}{V}\int_S (\mathbf{u}\cdot\mathbf{n})\,\phi \,\mathrm{d}S = 0, \tag{A.1}$$

where V is the cell volume, $\mathbf{n}$ is the outward unit normal, and S is the cell-face area. Discretizing the surface integral over all faces yields

$$\frac{\partial \phi}{\partial t} + \frac{1}{V}\sum_i u_{f,i}\,\phi_{f,i}\,S_i = 0. \tag{A.2}$$

Face values $u_{f,i}$ and $\phi_{f,i}$ are extrapolated from adjacent cells. A hybrid scheme blends upwind (stable but diffusive) and central differencing (accurate but less stable):

$$\phi_{f,i-\frac{1}{2}} = (1-\beta)\,\phi_{f,i-1} + \frac{\beta}{2}(\phi_{f,i-1} + \phi_{f,i})\,, \tag{A.3}$$

where β controls the upwind/central mix. Owing to hydrogen's high diffusivity and flashback risk, $\beta = 1$ (central) is used for all transported scalars except turbulence, which is treated fully upwind. Step limiters enforce monotonicity, reverting to first-order upwind when needed. Time integration uses an implicit first-order (backward Euler) scheme.

A.2. PISO Algorithm

The PISO method [64] couples pressure and velocity efficiently, allowing larger time steps with fewer iterations. Starting from a momentum predictor with interim values denoted by $(\cdot)^\star$ and previous-time fields by superscript n,

$$\frac{\rho^n\, u_i^\star}{\Delta t} - \frac{\rho^n\, u_i^n}{\Delta t} = -\frac{\partial P^n}{\partial x_i} + H_i^\star, \tag{A.4}$$

the velocity is corrected to satisfy mass conservation using an updated pressure $P^\star$,

$$\frac{\rho^\star\, u_i^{\star\star}}{\Delta t} - \frac{\rho^n\, u_i^n}{\Delta t} = -\frac{\partial P^\star}{\partial x_i} + H_i^\star. \tag{A.5}$$

Combining the predictor/corrector with continuity yields a pressure equation (for an ideal gas, $\rho^\star = P^\star/(Z^n R^n T^n)$ with $Z=1$):

$$\frac{\partial^2}{\partial x_i \partial x_i}(P^\star - P^n) - (P^\star - P^n)\,\frac{\phi^n}{\Delta t^2} = \left(\frac{\partial(\rho^n u_i^\star)}{\partial x_i} - S\right)\frac{1}{\Delta t}. \tag{A.6}$$

A second correction further refines u_i and P,

$$\frac{\rho^{\star\star}\, u_i^{\star\star\star}}{\Delta t} - \frac{\rho^n\, u_i^n}{\Delta t} = -\frac{\partial P^{\star\star}}{\partial x_i} + H_i^\star, \tag{A.7}$$

after which the pressure equation is updated and the corrected velocity is recomputed. Two correction loops typically suffice. Temperature and other scalars are corrected similarly. In this setup, the PISO loop uses a convergence multiplier of 20.0, enforces a minimum of two and a maximum of nine corrections, and stops when

$$\frac{|\Psi - \Psi_{t-1}|}{|\Psi^\star|} < \Psi_{\mathrm{tol}}, \qquad \Psi_{\mathrm{tol}} = 10^{-4}. \tag{A.8}$$

A.3. Rhie-Chow interpolation

To prevent pressure–velocity decoupling on collocated grids, Rhie–Chow interpolation [101] modifies face velocities during correction using pressure gradients over neighboring cells. In 1D notation (cell i, face $i+\frac{1}{2}$),

$$u^\star_{i+\frac{1}{2}} = u^\star_i + u^\star_{i+\frac{1}{2}} - \frac{\Delta t}{\rho}\left(\frac{P_{i+1}-P_i}{\Delta x}\right) + \frac{\Delta t^2}{\rho}\left(\frac{P_{i+1}-P_{i-\frac{1}{2}}}{2\,\Delta x} + \frac{P_{i+2}-P_i}{2\,\Delta x}\right), \tag{A.9}$$

which damps checkerboarding while preserving coupling.

A.4. Linear solver

At each iteration the discretized system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is solved with SOR. Convergence is monitored via the normalized residual

$$r^{(n)} = \frac{\|\mathbf{A}\mathbf{x}^{(n)} - \mathbf{b}\|_2}{\|\mathbf{b}\|_2}, \tag{A.10}$$

and over-relaxation with factor ω accelerates decay of $r^{(n)}$. The SOR update for component i is

$$x_i^{(k+1)} = (1-\omega)\,x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}\,x_j^{(k+1)} - \sum_{j>i} a_{ij}\,x_j^{(k)}\right). \tag{A.11}$$

B

Navier-Stokes Characteristic Boundary Condition

This method is derived from the eigenanalysis of the Euler equations, allowing pressure waves that match the far-field (upstream or downstream) conditions to pass through the boundaries, while reflecting others. In CONVERGE CFD, NSCBC implementation follows the approach outlined by Thompson [115], [116], and Poinsot and Lelef [94].

The three-dimensional Euler equations yield five characteristic eigenvalues, each corresponding to different wave types at the boundary:

$$\lambda_1 = u,$$
$$\lambda_2 = u,$$
$$\lambda_3 = u,$$
$$\lambda_4 = u - c,$$
$$\lambda_5 = u + c,$$

Here, λ_1 to λ_3 represent entropy and vorticity waves, while λ_4 and λ_5 correspond to acoustic waves. The wave propagation speed is given by λ, with direction $\frac{dn}{dt} = \lambda$.

At a subsonic inlet, four eigenvalues are positive (incoming) and one is negative (outgoing), necessitating four physical boundary conditions and one numerical condition. Conversely, at a subsonic outlet, four eigenvalues are negative (outgoing) and one is positive (incoming), requiring one physical condition and four numerical ones.

During the LES, to update the velocity, density, and pressure in each iteration, a correction-based NSCBC is employed. Its purpose is to utilize the local one-dimensional inviscid (LODI) formulation where if $U(\rho, p, u, v, w)$ would represent the state vector, the time derivative of its residual R would be,

$$\frac{\partial U}{\partial t} = -R \tag{B.1}$$

The residual given at the end of the PISO algorithm is,

$$R^P = -\frac{U^{n+1,P} - U^n}{dt} \tag{B.2}$$

Next, the correction-based NSCBC will adjust the incoming waves into corrected waves that match the boundary conditions that are imposed (C). By doing so, R is able to be split into $R = ML$, where M is a matrix system found in the work of Poinsot and Lelef [94], and L is a wave amplitude vector:

$$U^{n+1,C} = U^n - dt\left(R^P - R_{BC}^{in,P} + R_{BC}^{in,C}\right) \tag{B.3}$$

The incoming characteristic residuals are modeled as $R_{BC}^{in,P} = ML^{in}$ and $R_{BC}^{in,C} = ML^{in,C}$, where M is a transformation matrix. At the inlet, where four waves propagate into the domain, the wave amplitude vectors are defined as:

$$L^{in} = (0, L_2, L_3, L_4, L_5), \quad L^{in,C} = (0, L_2^C, L_3^C, L_4^C, L_5^C).$$

At the outlet, only one incoming wave is present, and the vectors become:

$$L^{in} = (L_1, 0, 0, 0, 0), \quad L^{in,C} = (L_1^C, 0, 0, 0, 0).$$

The corrected characteristic wave amplitudes L_i^C for the inlet are obtained from the following expressions:

$$\frac{\partial u}{\partial t} = -\frac{1}{2\rho c}L_5^C = -K(u - u_\infty), \tag{B.4}$$

$$\frac{\partial v}{\partial t} = -L_3^C = -K(v - v_\infty), \tag{B.5}$$

$$\frac{\partial w}{\partial t} = -L_4^C = -K(w - w_\infty), \tag{B.6}$$

$$\frac{\partial T}{\partial t} = -\frac{T}{\rho c^2}L_2^C = -K(T - T_\infty), \tag{B.7}$$

where $u_\infty, v_\infty, w_\infty$, and T_∞ denote the far-field (Dirichlet) values.

For the outlet, the incoming wave component is computed using:

$$L_1 = K(p - p_\infty), \tag{B.8}$$

with p_∞ representing the far-field pressure. The corrected amplitude L_1^C is then:

$$\frac{\partial p}{\partial t} = -\frac{1}{2}L_1^C = -K(p - p_\infty). \tag{B.9}$$

The relaxation coefficient K, used to enforce far-field conditions smoothly, is given by:

$$K = \sigma(1 - M^2)\frac{c}{L}, \tag{B.10}$$

In this formulation, M denotes the Mach number, σ is a user-defined tuning parameter, and L represents the characteristic length, set to 0.06 m in this work. The relaxation constant K plays a signifcant role in ensuring the accuracy and stability of the boundary treatment, making the appropriate selection of σ essential.

If σ is chosen too large, it may amplify wave magnitudes excessively, causing numerical instabilities or divergence as the velocity field diverges from the desired target. On the other hand, selecting a very low σ value results in boundaries that are nearly transparent to outgoing waves, but this can lead to a gradual deviation of the mean solution due to viscous and transverse effects inherent in the Navier-Stokes equations.

Therefore, an ideal σ must balance these effects—minimizing reflections while maintaining solution stability and accuracy. In this work, a value of $\sigma = 0.25$ is adopted, following the recommendation by Rudy and Strikwerda [107].

C

Spray Characteristics

This appendix chapter details the characteristics of the spray model.

C.1. Injection Size Distribution

The Rosin–Rammler distribution is used for this parameter. Since coalescence and break-up are neglected, a more complex, non-uniform distribution is applied to compensate. This approach strikes a balance between accuracy and the reduced computational cost. The cumulative probability distribution function in this context is,

$$p(r) = 1 - \exp\left(\zeta^{C_{RR}}\right), \quad 0 < \zeta < \zeta_{\text{max}} \tag{C.1}$$

In this equation, $\zeta = \frac{r}{\tilde{r}}$ with a maximum value of $\zeta_{max} = \ln 1000^{\frac{1}{C_{RR}}}$. This is done to limit the maximum radius, using the C_{RR} constant. Next,

$$\tilde{r} = \Gamma\left(1 - \frac{1}{C_{RR}}\right) r_{\text{Sauter}} \tag{C.2}$$

where r_{sauter} is the Sauter radius, and Γ is the Gamma function. Finally, the injected radius is $r = \tilde{r}\zeta$. Next, the particle dynamics will be explored.

C.2. Particle Dynamics

As a Lagrangian motion will be used for the path of the droplets, a deep understanding of the particle dynamics is needed. An initial spraying of the droplets resulting in an interaction of the fluid with gas, which is modeled by Newton's second law,

$$\rho_d V_d \frac{du_i}{dt} = F_{d,i} \tag{C.3}$$

where ρ_d is the droplet's density, V_d is droplet's volume, u_i is the velocity of droplet, and $F_{d,i}$ is the sum of all forces. The subscript i represents each droplet. It should be noted that pressure and body forces are neglected, and primarily drag is considered for simplicity, and is shown as

$$F_d = F_{\text{drag}} = \frac{1}{2} C_D A \rho_g u_{dg}^2 \tag{C.4}$$

where $u_{dg} = u_g + u_g^{'} - u_d$ and u_g is the gas velocity, $u_g^{'}$ is the turbulent gas fluctuations, and u_d is the droplet velocity. A is the frontal area, ρ_g is the density of the gas. Considering that the frontal area is a function of evaporation (breakup and coalescence as well had they not been ignored), the area is variable. Furthermore, as

$$V_d = \frac{4}{3}\pi r^3 \tag{C.5}$$

Equation C.3 can be rewritten as,

$$\frac{du_i}{dt} = \frac{3}{8}\frac{\rho_g}{\rho_d} C_D \frac{u_{dg}^2}{r} \tag{C.6}$$

Finally, C_D, the drag coefficient, can be modeled using an assumption that the droplet remains spherical. Then, according to Liu, Mather, and Reitz [79], it can be modeled as:

$$C_D = \begin{cases} \frac{24}{Re_d}\left(1 + \frac{1}{6}Re_d^{2/3}\right), & Re_d \leq 1000, \\ 0.424, & Re_d > 1000, \end{cases} \tag{3.37}$$

where Re_d is the Reynolds number of the droplet, depending on the gas properties, its diameter, and its relative velocity u_{dg}. Although this formulation of the drag coefficient is simple, this value is under-predicted at high Weber numbers, due to the droplet forming a disc-like geometry. This would lead to a higher drag value than is predicted, so Liu, Mather, and Reitz [79] proposes an extra equation,

$$C_D = C_{D,\text{sphere}}\left(1 + 2.632\, y\right) \tag{C.7}$$

where y is the drop distortion. At zero distortion, the drag coefficient remains as a sphere, however at 1, the drag of a disc is used. For under-damped drops,

$$y(t) = We_c + e^{-t/t_d}\left[(y(0) - We_c)\cos(\omega t) + \frac{1}{\omega}\left(\frac{dy}{dt}(0) + \frac{y(0) - We_c}{t_d}\right)\sin(\omega t)\right] \tag{C.8}$$

where,

$$We_g = \frac{\rho_g u_{rel}^2 r_o}{\sigma} \tag{C.9}$$

$$We_c = \frac{C_F}{C_k C_b} We_g \tag{C.10}$$

$$\frac{1}{t_d} = \frac{C_d}{2}\frac{\mu_l}{\rho_l r_o^2} \tag{C.11}$$

$$\omega^2 = C_k \frac{\sigma}{\rho_l r_o^3} - \frac{1}{t_d^2} \tag{C.12}$$

where C_k, C_F, and C_b are model constants; ω denotes the oscillation frequency; We_g is the Weber number of the droplet; u_{rel} represents the velocity relative to the local flow; σ is the droplet surface tension; μ_l is the liquid viscosity; and r_0 is the radius of the droplet in its undisturbed state. Furthermore, a turbulent dispersion model is needed to provide accuracy.

C.3. Turbulent Dispersion

Turbulence strongly influences droplet motion, making it challenging to predict particle dispersion. Drag forces cause droplets to decelerate, transferring their momentum to the surrounding fluid at smaller scales. In LES analyses, accurately estimating velocities at these subgrid scales is essential and can

be approached using a simple Taylor expansion,

$$u_{\text{sub},i} = C_{\text{les}} \frac{dx^2}{24} \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} \tag{C.13}$$

where d is a characteristic cell dimension obtained by the cube root of the volume of the cell. C_{les} is a pre-determined constant, and $\bar{u}_i$ is the average velocity. The turbulent parts of the velocity are selected in intervals of the turbulence characteristic time, t_d. This value is the lesser of the time it takes for the droplet to pass an eddy, and the time for the eddy to dissolve,

$$t_d = \min\left(\frac{k_{\text{sg}}}{\varepsilon_{\text{sg}}},\ c_{\text{ps}} \frac{k_{\text{sg}}^{3/2}}{\varepsilon_{\text{sg}}} \frac{1}{|u_i + u_i' - u_{\text{dg},i}|}\right) \tag{C.14}$$

where c_{ps} is an empirical constant, k_{sg} is the turbulent kinetic energy, ϵ_{sg} is the turbulent dissipation rate and is written as,

$$k_{\text{sub}} = \frac{1}{2} u_{\text{sg},i}^2 \tag{C.15}$$

$$\varepsilon_{\text{sg}} = \frac{k_{\text{sub}}^{3/2}}{d} \tag{C.16}$$

Next, the droplet's evaporation will be modeled.

C.3.1. Evaporation

Due to the immense heat in the combustion settings, this naturally causes the droplets to evaporate either fully or partially. This is reflected in their radius, which is variable during the LES. Furthermore, their influence on the temperature of the flow, and their convective effects coupled with mass diffusion make the modeling of the droplet evaporation tricky. The ratio of convective heat transfer and mass diffusion is called the Sherwood number, Sh_d. The evolution of drop radius can be modeled using the equation,

$$\frac{dr_0}{dt} = -\frac{\alpha_{\text{spray}} \rho_g D}{2\rho_l r_0} B_d\, Sh_d \tag{C.17}$$

where α_{spray} is the mass transfer coefficient scaling factor, $\frac{\rho_g}{\rho_l}$ a ratio of the gas to liquid density, D is the liquid-air mass diffusivity at temperature $\hat{T} = (T_g - 2T_d)/3$, and Sherwood number [8]. It also includes the Spalding mass transfer number, B_d, which is a normalized ratio of water vapor mass fraction Y_1^* to overall vapor mass fraction, Y_1. The Spalding mass transfer number and Sherwood number are thus found as,

$$B_d = \frac{Y_1^* - Y_1}{1 - Y_1^*} \tag{C.18}$$

$$Sh_d = \left(2.0 + 0.6\, Re_d^{1/2}\, Sc^{1/3}\right) \frac{\ln(1 + B_d)}{B_d} \tag{C.19}$$

The Re_d and Schmidt (Sc) numbers can be found using these formulas,

$$Re_d = \frac{\rho_{\text{gas}} \left|u_i + u_i' - u_{dg}'\right| d}{\mu_{\text{air}}} \tag{C.20}$$

$$Sc = \frac{\mu_{\text{air}}}{1.293\, D_0 \left(\frac{\hat{T}}{273}\right)^{n_0 - 1}} \tag{C.21}$$

where D_0 and n_0 are found experimentally. For Equation C.18, the water vapor mass fraction is found as,

$$Y_1^* = \frac{W_{C_nH_{2m}}}{W_{C_nH_{2m}} + W_{\text{mix}}\left(\frac{p_g}{p_v} - 1\right)} \tag{C.22}$$

where W is the molecular weight, and p_g is the gas pressure, and p_v is the vapor pressure of the droplet. Once the evolution is determined, the heat transfer attributes must also be evaluated. For droplets that are smaller than a certain radius,

$$\bar{A}_d Q_d = c_l m_d^\star \frac{dT_d}{dt} - \frac{dm_d}{dt} H_{vap} \tag{C.23}$$

will be used, where A_d is the droplet surface area, c_l is the liquid specific heat, m_d is the droplet mass, and H_{vap} is the latent heat taken at the droplet conditions. Furthermore, Q_d, the heat conduction to the droplet surface area is found using the Ranz-Marshall correlation, assuming only that conduction takes place. This is given by,

$$Q_d = \frac{\beta_{\text{spray}}\, Nu_d\, k_{\text{air}}\, (T_{\text{gas}} - T_d)}{\bar{d}_0} \tag{C.24}$$

where β_{spray} denotes the scaling factor for the heat transfer coefficient, k_{air} is the thermal conductivity evaluated at $\hat{T}$, and Nu_d represents the droplet Nusselt number. The Nusselt number is obtained from,

$$Nu_d = \left(2.0 + 0.6 Re_d^{1/2} Pr_d^{1/3}\right) \frac{\ln(1 + B_d)}{B_d} \tag{C.25}$$

which also incorporates the droplet Prandtl number, Pr_d, evaluated at $\hat{T}$. Here, the Nusselt number is determined similar to the Sherwood number in Equation C.19, instead using the Prandtl number,

$$Pr_d = \frac{\mu_{\text{gas}}(\hat{T})\, c_p(\hat{T})}{K_{\text{gas}}(\hat{T})} \tag{C.26}$$

where K_{gas} is a modeling constant. For droplets with a radius greater than a certain threshold, a more comprehensive effective thermal conductivity model developed by Abramzon and Sirigano [2] is used, accounting for spherically symmetric temperature distribution and potential recirculation effects. The governing heat transfer equation for the droplet is given by

$$\rho c_p \frac{\partial T}{\partial t} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(k_{\text{eff}} r^2 \frac{\partial T}{\partial r} \right) \tag{C.27}$$

where r is the radial distance from the droplet center. At the droplet surface, the boundary condition is expressed as

$$k_{\text{eff}} \left. \frac{\partial T}{\partial r} \right|_{r=R_d} = h\,(T_g - T(R_d, t)) + \rho L \frac{dR_d}{dt} \tag{C.28}$$

where h is the droplet–gas convection coefficient, T_g is the gas temperature, $R_d(t)$ is the droplet radius, c_p is the droplet specific heat, L is the latent heat of evaporation, k is the droplet thermal conductivity, and $T(R_d, t)$ is the droplet surface temperature. Furthermore, if recirculation effects are considered, the effective thermal conductivity k_{eff} replaces k, with $k_{\text{eff}} = \chi k$, where the enhancement factor χ is given by

$$\chi = 1.86 = 0.86\, \tanh\left(2.225 \log_{10}\left(0.03333\, Pe_d\right)\right) \tag{C.29}$$

where Pe_d is the Peclet number of the droplet. Despite most droplets evaporating before reaching the wall, this interaction must also be considered.

C.4. Drop-Wall Interaction

The occurence of the droplet reaching the wall is a rare but important interaction that must be accounted for. This model is based on the formulations of Naber and Reitz [89] and Manuel et al. [84], focusing on the angled impingement of liquid jets on a solid surface. It employs a three-dimensional empirical framework that enforces both mass and momentum conservation. In this formulation, the velocity component parallel to the wall remains unchanged, while the perpendicular component plays a decisive role in determining the nature of the impact. The model categorizes the impact behavior into two distinct regimes, defined by the Weber number (We) at the instant of collision:

$$We_i = \frac{\rho_l V_n^2 d_0}{\sigma} \tag{C.30}$$

where V_n is the velocity perpendicular to the surface. If the We number is less than 80, the droplet bounces back in an elastic manner, where the outgoing normal velocity is,

$$V_{n,o} = V_{n,i}\sqrt{\frac{We_o}{We_i}} \tag{C.31}$$

Here, the Weber number of the droplet that bounces back (We_o is,

$$We_o = 0.678\, We_i\, \exp\left(-0.04415\, We_i\right) \tag{C.32}$$

This is an empirical law based on the observations of Wachters and Westerling [124]. When the impinging Weber number is above 80, the jet model is applied. Therefore, the leaving droplet is analagous to a liquid jet leaving tangent to the surface. This sheet thickness that follows from the impinging jet is,

$$h(\psi) = h_\pi e^{\beta(1-\psi/\pi)} \tag{C.33}$$

where h_π is the sheet height when the droplet hits the wall perpendicularly. The sheet's thickness depends on the impingement angle, the β modeling parameter, and the angle at which the droplet leaves the surface. The equations for these are,

$$\sin\alpha = \left(\frac{e^\beta + 1}{e^\beta - 1}\right)\frac{1}{1 + \left(\frac{\pi}{\beta}\right)^2} \tag{C.34}$$

$$\psi = \frac{\pi}{\beta}\ln\left[1 - n\left(1 - e^{-\beta}\right)\right] \tag{C.35}$$

where n is a random number between 0 and 1. Finally, the collisions can be modeled.

C.5. Collisions

The selected collision model is the NTC approach introduced by Schmidt and Rutland [110], which is based on stochastic sub-sampling of droplet parcels within each computational cell. The algorithm begins by grouping parcels that occupy the same cell, after which a randomized subset of all possible parcel pairs is selected. The total number of collisions occurring over a time interval Δt is then calculated by summing the probabilities of all sampled collision pairs.

$$M_{\text{coll}} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{V_{i,j}\sigma_{i,j}\Delta t}{\forall} \tag{C.36}$$

where N is the number of droplets, and $\sigma_{i,j} = \pi(r_i + r_j)^2$ calculates the cross-sectional area of the collision. To more accurately account for collision probabilities, the term $qV\sigma$ is introduced, where q represents the number of droplets in a parcel and N_p denotes the total number of parcels within a computational cell. Incorporating this factor, the refined expression for estimating the number of collisions is given by,

$$M_{\text{cand}} = \frac{N_p^2 (qV\sigma)_{\text{max}} \Delta t}{2\forall} \tag{C.37}$$

and

$$M_{\text{coll}} = \sum_{i=1}^{\sqrt{M_{\text{cand}}}} q_i \sum_{j=1}^{\sqrt{M_{\text{cand}}}} \frac{q_j V_{i,j} \sigma_{i,j}}{(qV\sigma)_{\text{max}}} \tag{C.38}$$

A selected subset of parcels is employed to approximate the entire population, resulting in a method that is considerably faster than other approaches with comparable accuracy. This outcome is then used to identify candidate pairs of parcels that may be involved in potential collisions. The presence of an actual collision between candidate parcels i and j is subsequently verified by:

$$r < \frac{q_j V_{i,j} \sigma_{i,j}}{(qV\sigma)_{\text{max}}} \tag{C.39}$$

where r is a random float between 0 and 1, and q_g indicates the drop count in the path of the collisions. Once this condition is satisfied, a collision is initiated. The outcome of the collision, ranging from mere contact to coalescence, deformation, or rebound, is governed by the model proposed by Post and Abraham [97], which classifies the interaction based on the Weber number. For more intricate behaviors, the criteria outlined by Ashgriz and Poo [11] as well as by Hou and Schmidt [58] are employed to distinguish among merging, stretching, or separation phenomena.

D

2 Latent Variable Analysis

This appendix documents the $d = 2$ bottleneck, included to illustrate the representational limits of too few latent variables. Comparisons are made primarily against the chosen $d = 3$ model.

Table D.1: Summary of hyperparameters explored for 2 latent variables

Hyperparameter	Search Range / Options	Best value
Hidden layer widths	$(12, 6)$	
L1 regularisation weight	8.989×10^{-6}	
L2 regularisation weight	2.137×10^{-5}	
Encoder activation	`linear`	
Output activation	`sigmoid`	
Latent activity L1	3.146×10^{-6}	
Learning rate	8.914×10^{-4}	
Batch size	16	

The 2-latent model converged to a shallow, near-linear architecture with minimal regularisation. This implies a compact, low-capacity code.

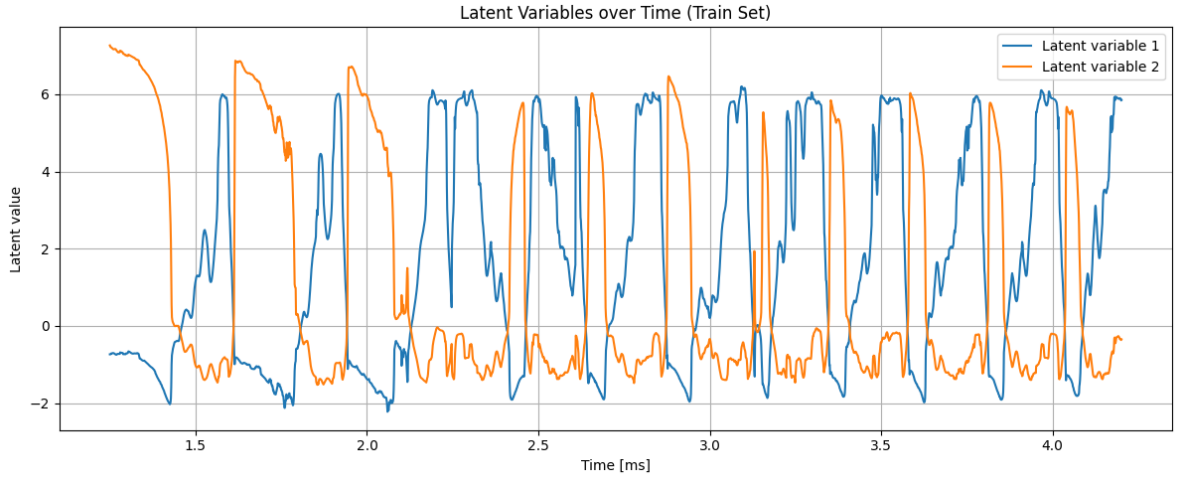


Figure D.1: 2 Latent Variables Visualized

Figure D.1 shows one coordinate acting as a quasi-binary state variable (on/off phases) and the other encoding continuous intensity/shape. Together they form a thin loop in 2D latent space—cycle-locked but highly compressed.

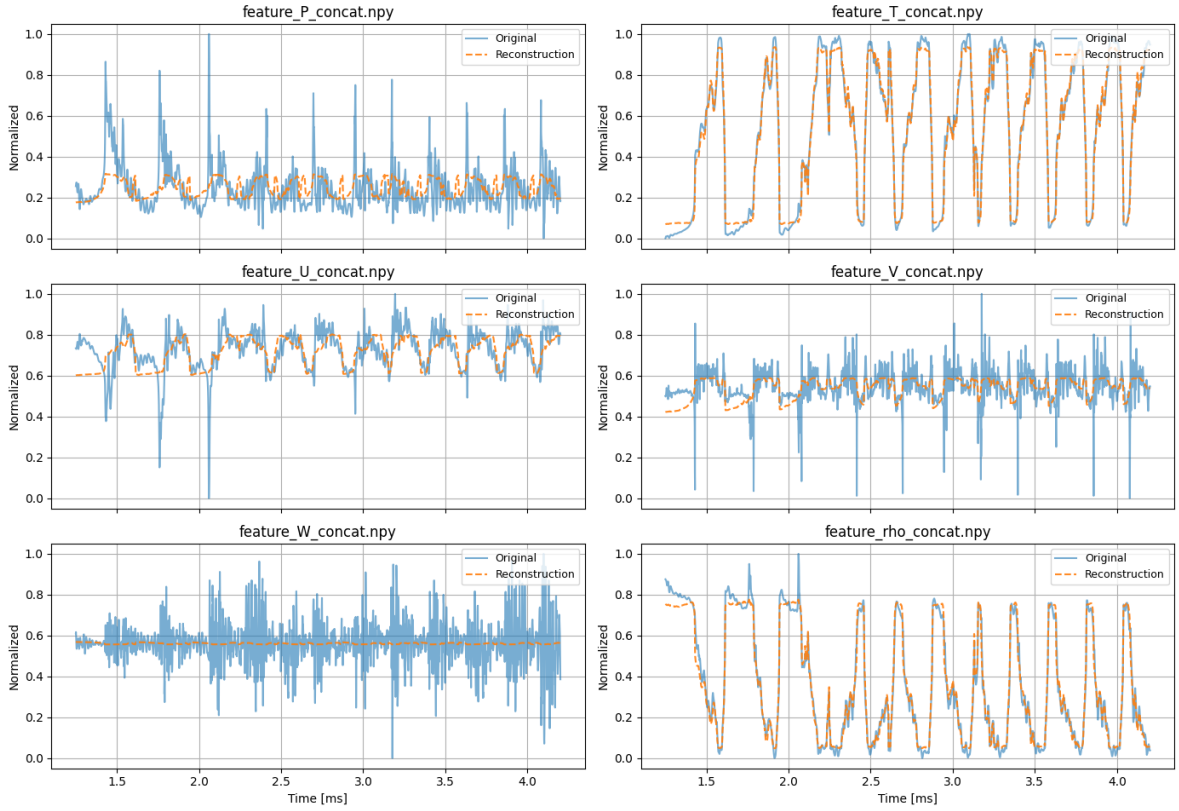


Figure D.2: Reconstruction of Thermodynamic and Velocity Features ($d = 2$)

Reconstructions (Figure D.2) are strong for cyclic, high-SNR variables (T, ρ) but systematically under-represent high-frequency dynamics in P, u, v, w. With only two latents, variance is allocated to dominant modes, leaving sharp transients smoothed out.

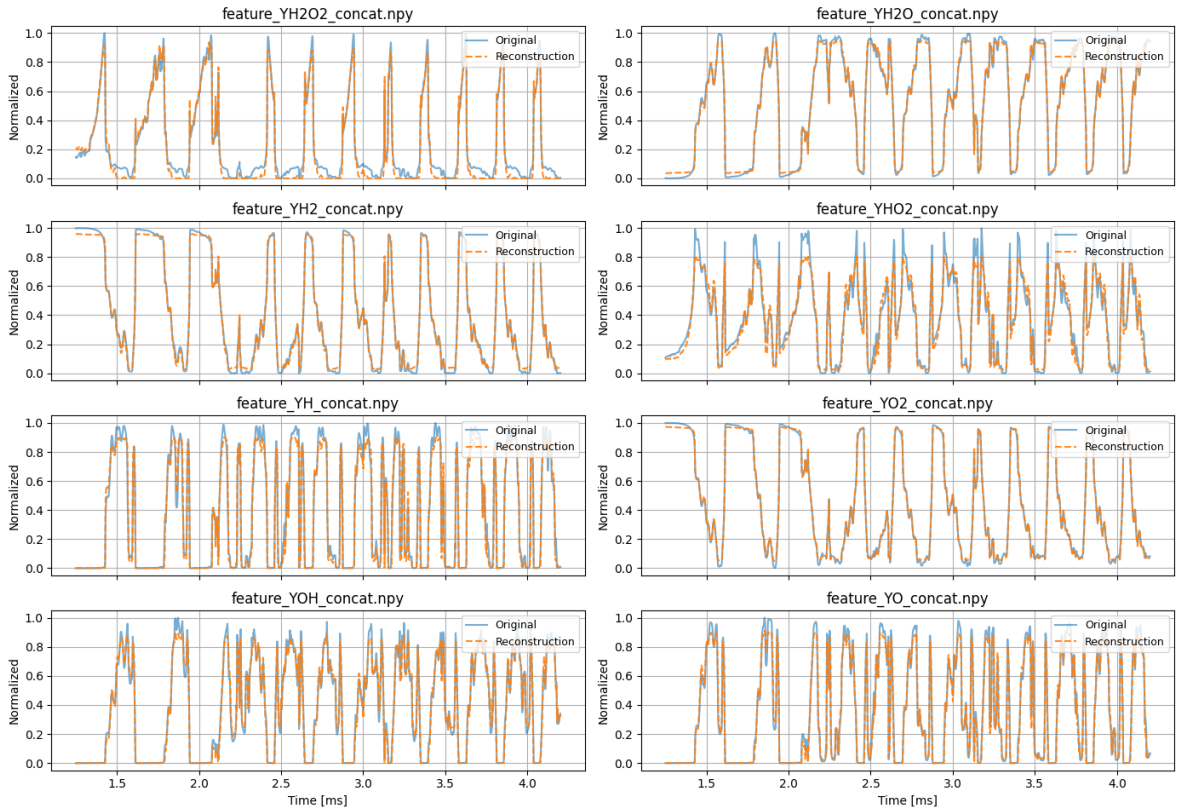


Figure D.3: Species Reconstructions ($d = 2$)

Species reconstructions (Figure D.3) show the same pattern: bulk species at ceiling, radicals/intermediates smoothed. The test set (**??**, **??**) preserves this ranking, with slightly more smoothing and small phase lags at sharp ramps.

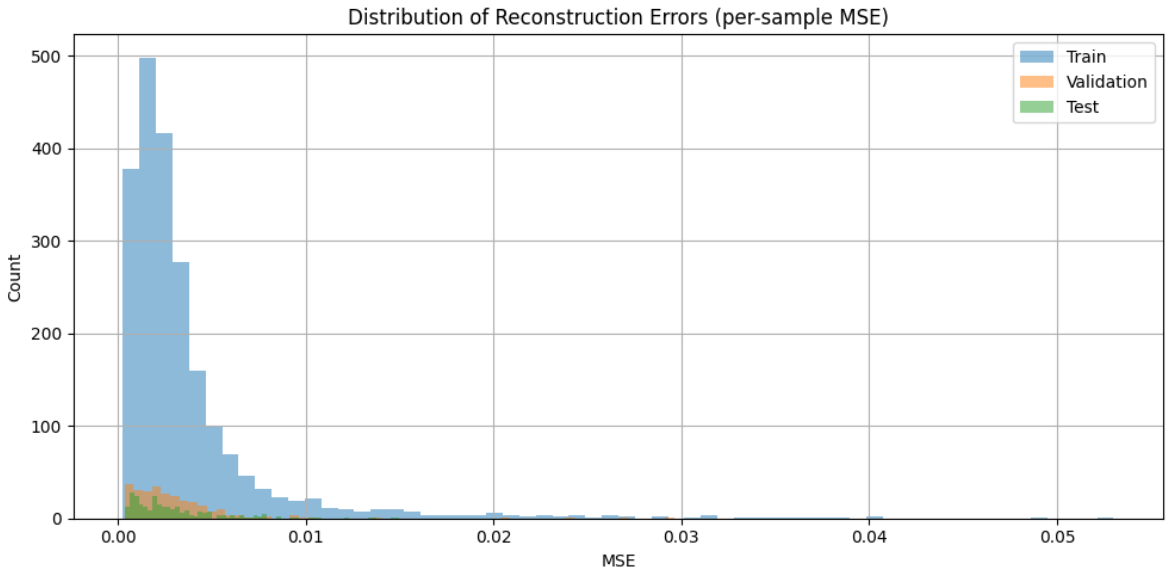


Figure D.4: MSE Distribution ($d = 2$)

The MSE histogram (Figure D.4) is broad and right-skewed, with a heavy tail from transient-rich windows. Loss evolution (Figure D.5) plateaus near 4×10^{-3}, reflecting the representational ceiling of a 2D code.

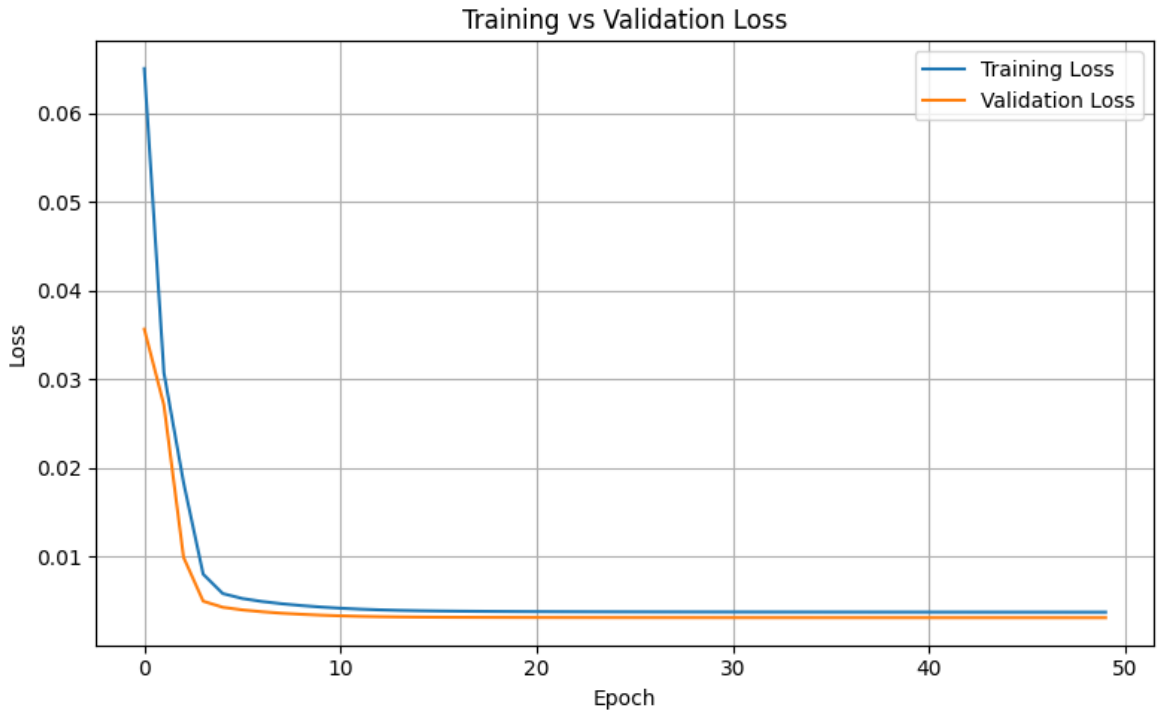


Figure D.5: Loss of 2 Latent Variables

The takeaway versus $d = 3$ is that two latents capture cycle phase and intensity but discard much of the mid/high-frequency structure. Compared with the 3-latent model, the reconstructions are visibly smoother, losses higher, and test generalisation weaker. This configuration demonstrates the lower bound of representational adequacy and motivates the need for a third latent.

E

4 Latent Variable Analysis

This appendix documents the $d = 4$ bottleneck, included to illustrate the effect of adding capacity beyond the chosen $d = 3$ model. Comparisons are made primarily against $d = 3$.

Table E.1: Summary of hyperparameters explored for 4 latent variables

Hyperparameter	Search Range / Options	Best value
Hidden layer widths	$(10, 8, 6)$	
L1 regularisation weight	8.327×10^{-6}	
L2 regularisation weight	4.340×10^{-5}	
Encoder activation	`linear`	
Output activation	`sigmoid`	
Latent activity L1	5.000×10^{-5}	
Learning rate	9.795×10^{-4}	
Batch size	16	

The 4-latent model retains a near-linear architecture but introduces strong regularisation, particularly a heavy latent L1 penalty, encouraging sparsity.

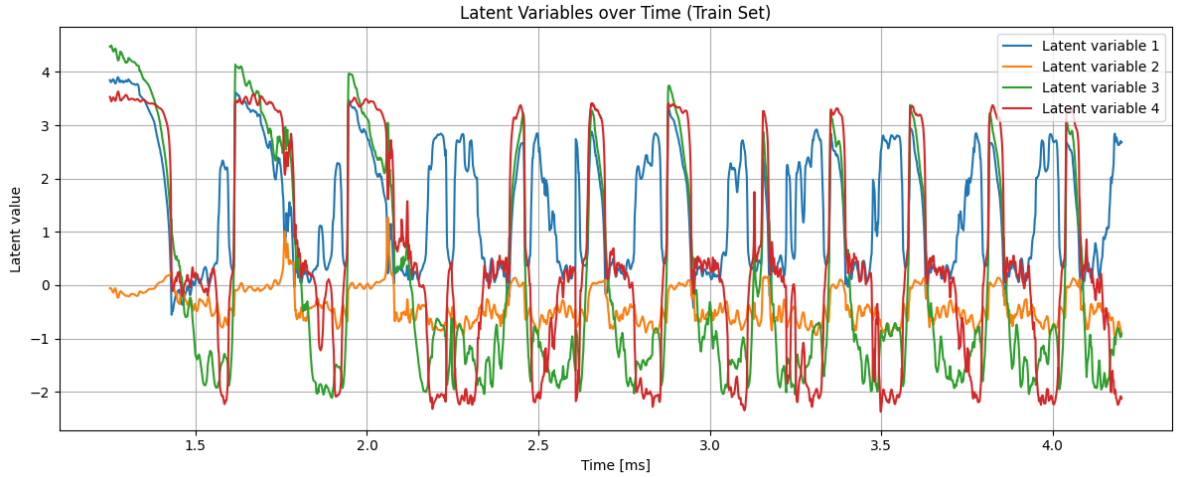


Figure E.1: 4 Latent Variables Visualized

Figure E.1 shows clearer role division: (i) baseline offset, (ii) smooth intra-cycle modulation, (iii) state/intensity, (iv) a sparse transition detector. Relative to $d = 3$, the fourth coordinate mainly isolates short-lived ramps, reducing redundancy.

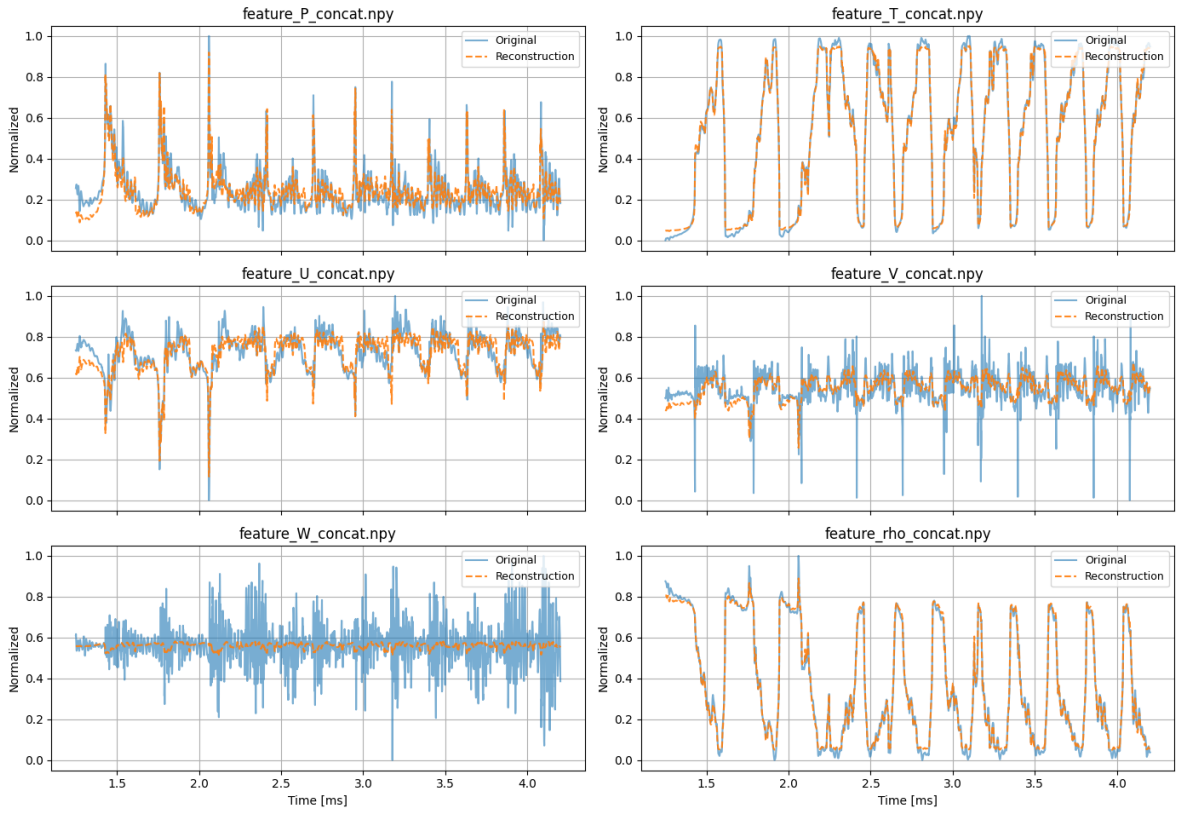


Figure E.2: Reconstruction of Thermodynamic and Velocity Features ($d = 4$)

Reconstructions (Figure E.2) remain ceiling-level for T, ρ; gains appear in P, u, v, w, with reduced amplitude bias and lag. Compared with $d = 3$, the improvement is modest—slightly sharper envelopes, but qualitatively similar.

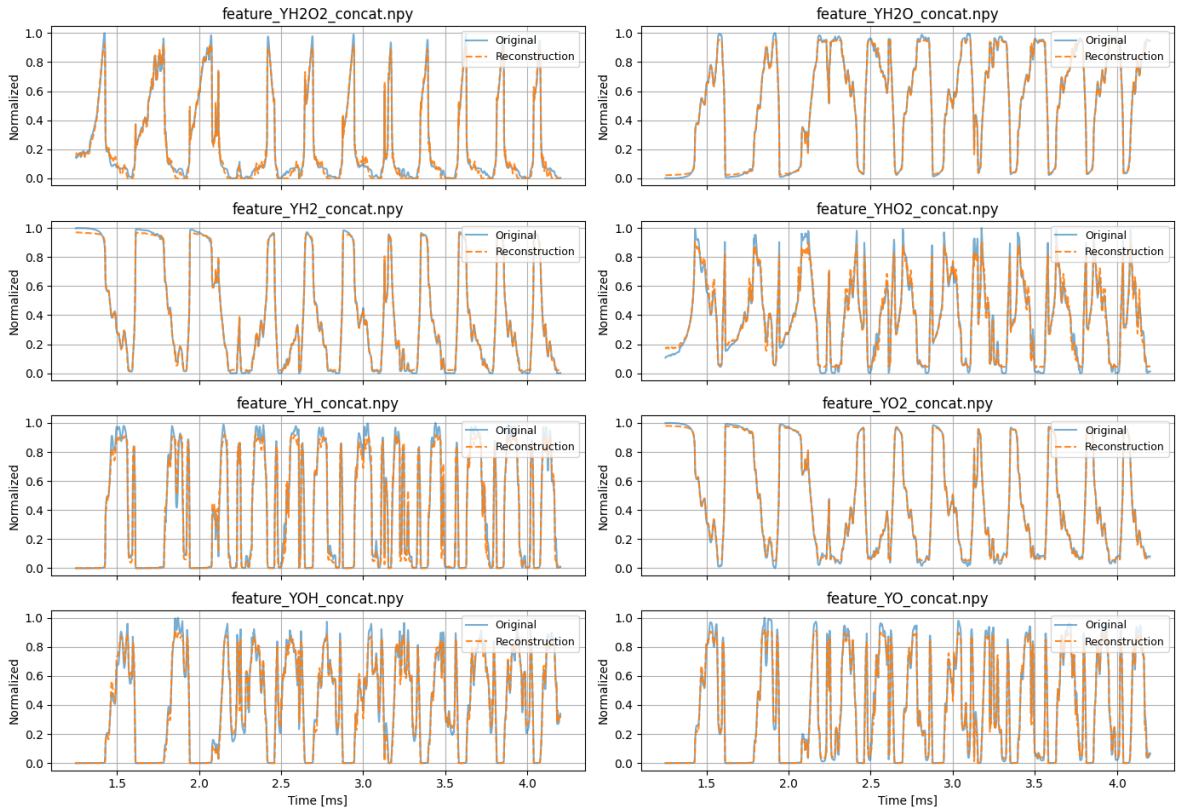


Figure E.3: Species Reconstructions ($d = 4$)

Species reconstructions (Figure E.3) are uniformly excellent. Radicals/intermediates benefit marginally: peaks are a touch sharper, troughs less filled. Relative to $d = 3$, differences are incremental, not transformative.

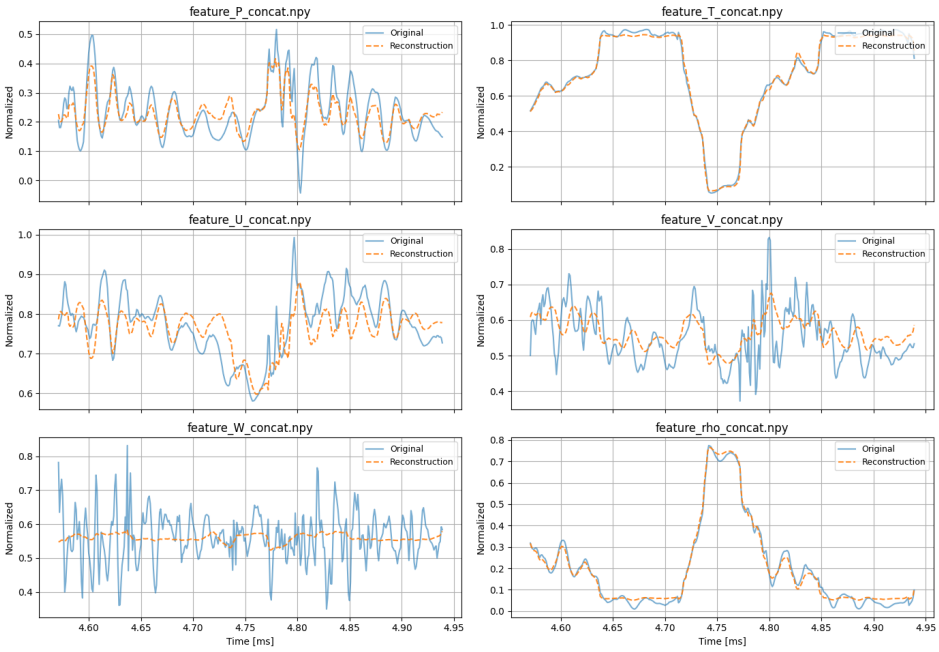


Figure E.4: Thermodynamic/velocity reconstructions ($d = 4$, test)

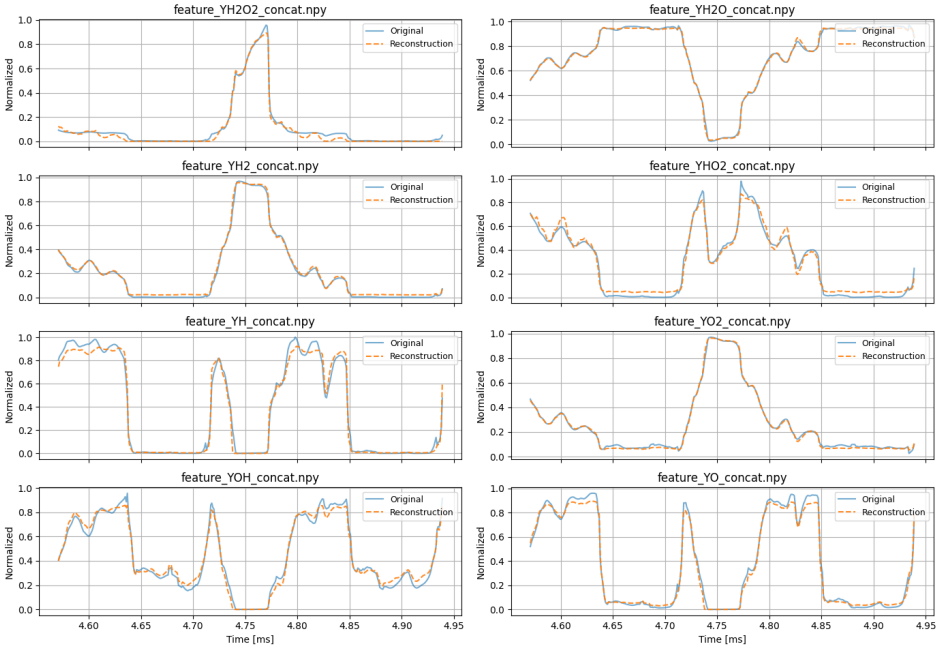


Figure E.5: Species reconstructions ($d = 4$, test)

Test results confirm this: cyclic variables unchanged, noisy channels modestly improved, radicals slightly sharper. Again, the step from 2→3 yields the largest benefit; 4 is a refinement.

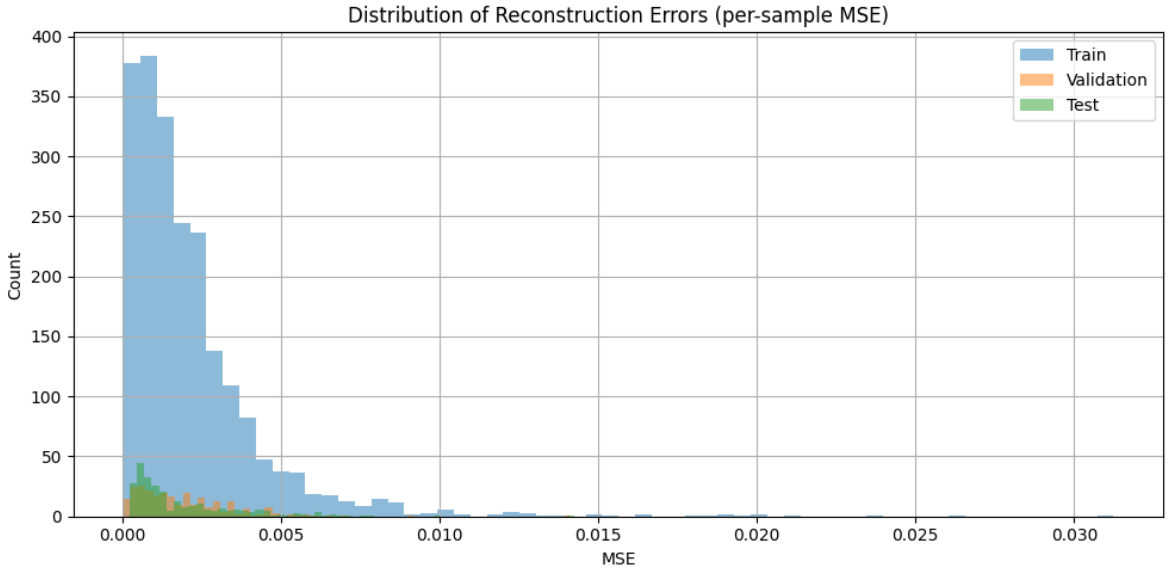


Figure E.6: MSE Distribution ($d = 4$)

MSE distributions (Figure E.6) shift further left compared with $d = 3$ (Figure 7.19), with fewer tail outliers. Loss evolution (Figure E.7) mirrors this: a lower asymptotic level than $d = 3$, but with slower convergence.

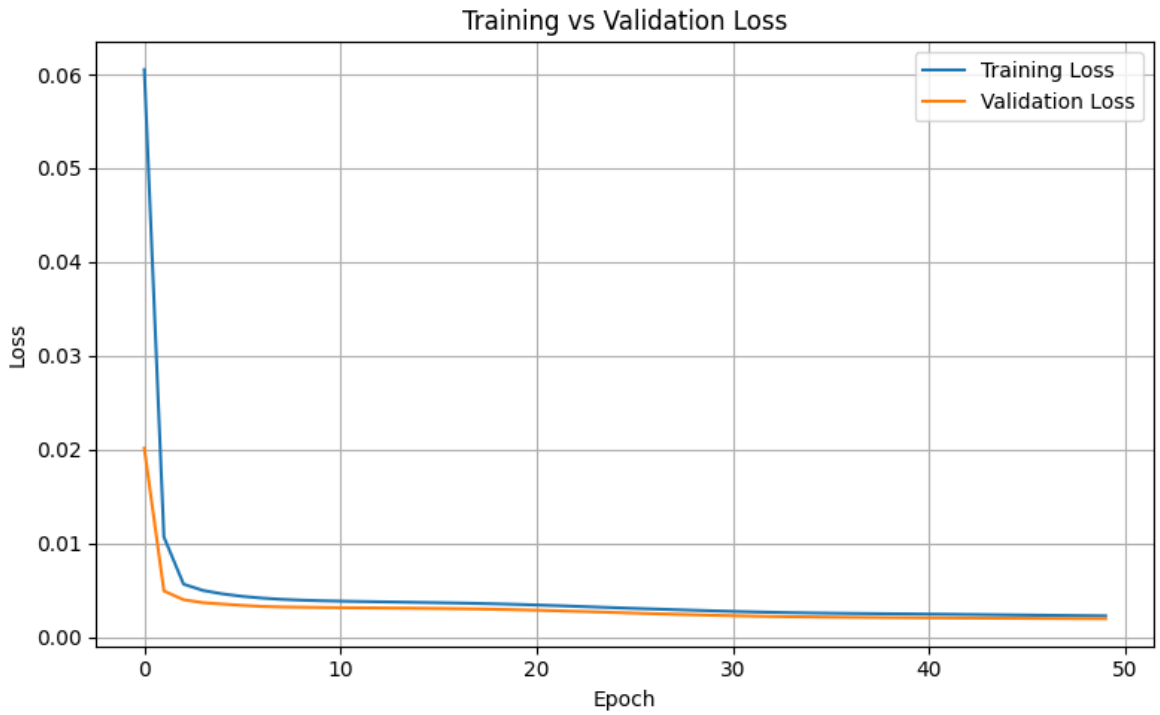


Figure E.7: Loss of 4 Latent Variables

Overall, 4 latents improve reconstructions and reduce error variance, especially for transients, but only incrementally relative to three. The gains are consistent but small, reflecting diminishing returns. The

4D bottleneck therefore validates the 3D choice as a near-optimal operating point: compact, expressive, and sufficient.