# Evaluating Novel Methods of Data Post-Processing in Computational Fluid Mechanics

Antoni Kowalik
antonikowalik23@gmail.com

under the direction of
Prof. Christophe Duwig
Department of Chemical Engineering
Royal Institute of Technology

Research Academy for Young Scientists
July 13, 2022

## Abstract

High-dimensional data is ubiquitous is Computational Fluid Dynamics and many other fields of science, presenting unique difficulties in analysis, due to phenomena known as the "curse of dimensionality", the visual way in which humans interpret data, and the complex relationships found in high-dimensional data sets. To address these issues, a novel workflow based on modern data-driven algorithms was developed by Rovira et al. [1]. In the present work, this workflow is applied to a new reactive flow data set from a study by Zhang et al. [2], and its performance, applicability, and consistency over the time evolution of the system is evaluated. The workflow was used to automatically identify and describe clusters of data with similar physical properties in a logical, explainable and useful way. It was found to be consistent, simple to understand and apply, and potentially adaptable to a wide range of use cases. The various results obtained using the workflow are analysed, and certain special cases and limitations are discussed. New possible applications of the workflow and future research directions are proposed.

# Acknowledgements

I would like to thank Prof. Christophe Duwig for his guidance and the opportunity to do this project. I would also like to express my gratitude towards Dr. Marc Rovira for taking all the time I needed to introduce me to this field and his work in it, and for coming to the rescue when things went wrong. I also want to thank Dr. Kai Zhang for introducing me to the basics of computer simulation in CFD, the data set used in this work, and the physics involved. I am also grateful to Mikael Häglund for his companionship during our two weeks in a dark, hot office with no functional A/C, and for his compassion when my code did what I told it to do instead of what I wanted it to do. Finally, I want to thank Rays — for excellence, and their partners Beijerstiftelsen and Olle Engkvist Stiftelse, for sending me on this journey in the first place.

# Contents

# 1 Introduction

Computer simulations generating large amounts of data are common in both scientific research and industrial applications [3]. They allow researchers and engineers to be less reliant on experimental testing, reducing cost and complexity of research. One field to which numerical simulation is fundamental is Computational Fluid Dynamics (CFD), where computer simulations are employed to better understand fluid flow. [4]

## 1.1 High-Dimensional Data in Computational Fluid Dynamics

CFD simulations often produce a large number of data points in many degrees of freedom. Degrees of freedom refers to the number of independent features each point is described by, the variables of each point, or in other words the number of measurable quantities each data point in a data set contains. These variables can be viewed as coordinates in a space the dimension of which is the number of variables. For example, in a three dimensional space each point can be described by three coordinates, or variables, while in a 20 dimensional space 20 variables are required. In CFD, such variables typically include the velocity of the fluid at each point, density, temperature, and chemical composition of the fluid, as well as other features specific to each simulation. Typical CFD data sets contain millions of data points in tens or sometimes hundreds of dimensions. Such high-dimensional data sets, while containing a lot of information about the simulated systems, are difficult to analyse. [5]

Humans rely primarily on visual aids such as plots and diagrams to identify relationships in data, which limits us to analysing two or three variables at a time. The number of plots required to find relationships between a higher number of variables makes this method of analysis infeasible for high-dimensional data. Data analysis, even by non-manual methods, is further complicated by the the complexity of phenomena involved in fluid flow, such as turbulence, which are not generally well-understood. Additionally, CFD data sets commonly exhibit complex, non-linear correlations between many variables [6]. This limits the use of techniques for analysing high-dimensional data such as Principal Component Analysis, PCA, which relies on matrix multiplication and can thus only capture linear relationships.

A multi-step workflow aiming to overcome these problems, proposed by Fooladgar and Duwig [5], involves dimensionality reduction, clustering, and feature correlation. These methods aim to extract the key features of data for easier visualisation, or as input for further analysis. This workflow was further developed by Rovira, Engvall, and Duwig [1], utilising more modern algorithms and machine-learning models. The method employs the Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) algorithm for dimensionality reduction, Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) for clustering, and Mutual Information (MI) for feature correlation. The workflow is visualised in Figure 1.

## 1.2 Dimensionality Reduction

Dimensionality reduction is a family of methods for embedding a high-dimensional data set into usually two or three dimensions such that similar and dissimilar points are represented by nearby and distant points, respectively. In other words, dimensionality reduction methods aim to reduce the number of dimensions of a data set while preserving the structure and relationships between data points. There are two main reasons for doing this. Firstly, it presents the data in a lower dimensional form which can be analysed manually. Secondly, it is a method to structure the data as input for other analysis steps. Many data analysis methods suffer from difficulties in high-dimensional spaces [7]. This is particularily noticable in clustering, described in section 1.3. These phenomena are often referred to as the “curse of dimensionality“. As dimensionality increases the volume of the space, and thus the number of points needed to maintain density, increases exponentially. In many degrees of freedom even large data sets appear sparse and distinction between dissimilar and similar points diminishes. Embedding the data into a lower dimensional space allows similarities and differences between points to be more apparent.

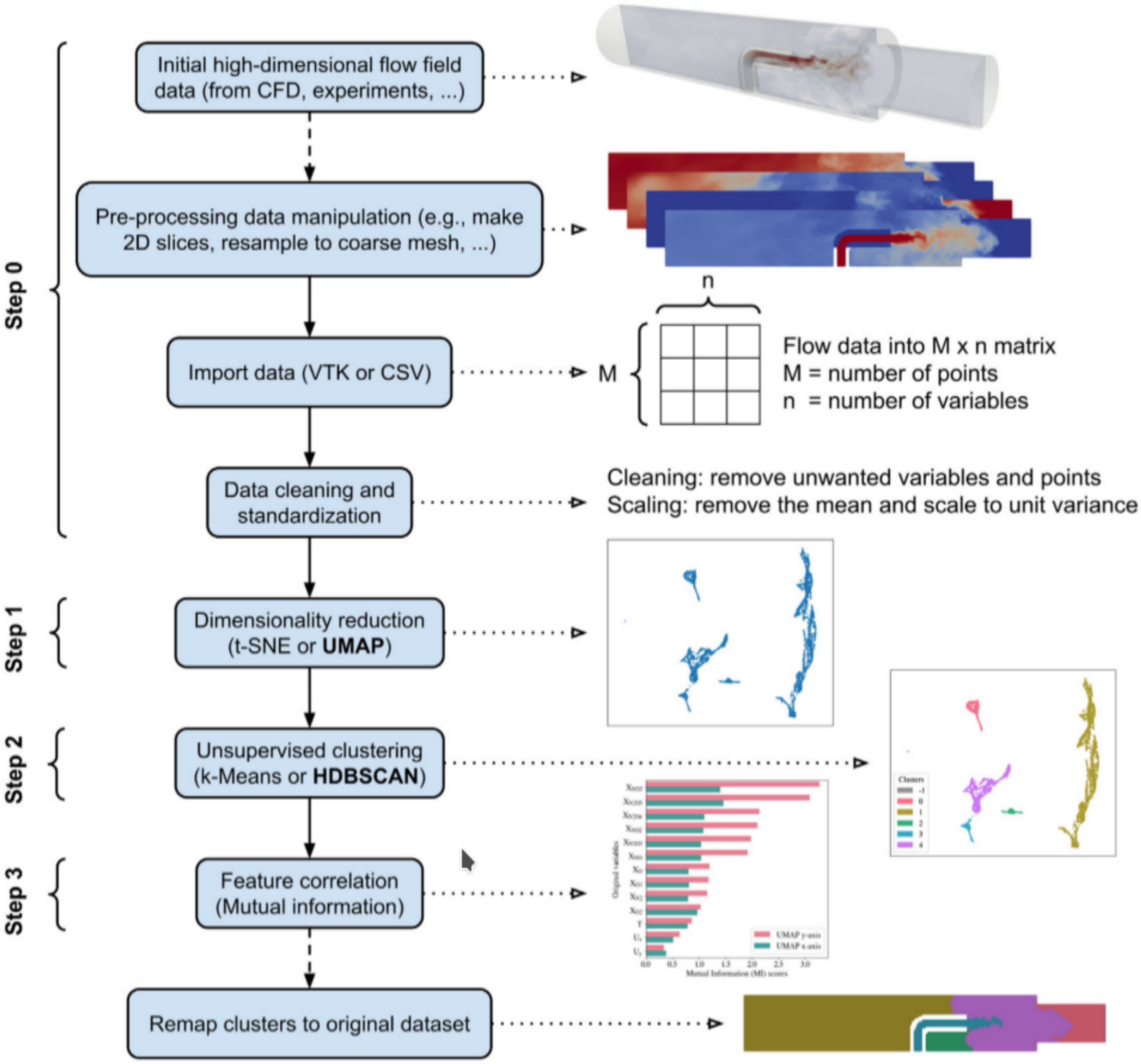

Step 0
Initial high-dimensional flow field data (from CFD, experiments, ...)
Pre-processing data manipulation (e.g., make 2D slices, resample to coarse mesh, ...)
Import data (VTK or CSV)
n
M
Flow data into M x n matrix
M = number of points
n = number of variables
Data cleaning and standardization
Cleaning: remove unwanted variables and points
Scaling: remove the mean and scale to unit variance
Step 1
Dimensionality reduction (t-SNE or UMAP)
Step 2
Unsupervised clustering (k-Means or HDBSCAN)
Clusters
-1
0
1
2
3
4
Step 3
Feature correlation (Mutual information)
Original variables
UMAP y-axis
UMAP x-axis
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Mutual Information (MI) scores
Remap clusters to original dataset

Figure 1: Flowchart of workflow proposed by Rovira et al. [1], from the original paper.

Dimensionality reduction aims to reduce high-dimensional data into groups of similar points. This is helpful in finding areas in the data with similar physical properties. However, dimensionality reduction by it self does not identify such groups. Additionally, coordinates of each point in the lower-dimensional embedding, commonly called "synthetic variables", hold no physical significance, unlike variables in the original high-dimensional space. Identification of the physical properties of groups in the embedding thus requires add the back original variables to the dimensionality reduced data, adding back complexity into the analysis [1]. These problems are dealt with in the next steps of the workflow, in which clusters of similar points in the embedded data are identified, and the synthetic variables are correlated with the original variables for each cluster.

## 1.3 Clustering

Clustering algorithms classify data into groups of similar points, called clusters. A simple example of clustering is shown in Figure 2. When applied to CFD data sets, the aim of these algorithms is to identify regions of points with similar physical properties and processes. While humans could potentially identify similar regions in high-dimensional data embedded into lower-dimensional spaces themselves, clustering algorithms automate this process, allowing for analysis of large quantities of data. Additionally, in complex data sets the separation between distinct groups is often not clear-cut. Clustering algorithms can often identify patterns and relationships that are hard for humans to notice, while providing clear, mathematically substantiated motivation.

## 1.4 Feature Correlation

The remaining problem, solved by feature correlation, is that the physical properties of the clusters still have not been identified. Feature correlation refers to correlating the synthetic variables of the embedding with the variables of the original data set, to key features and common properties of each cluster. The results obtained from this step and

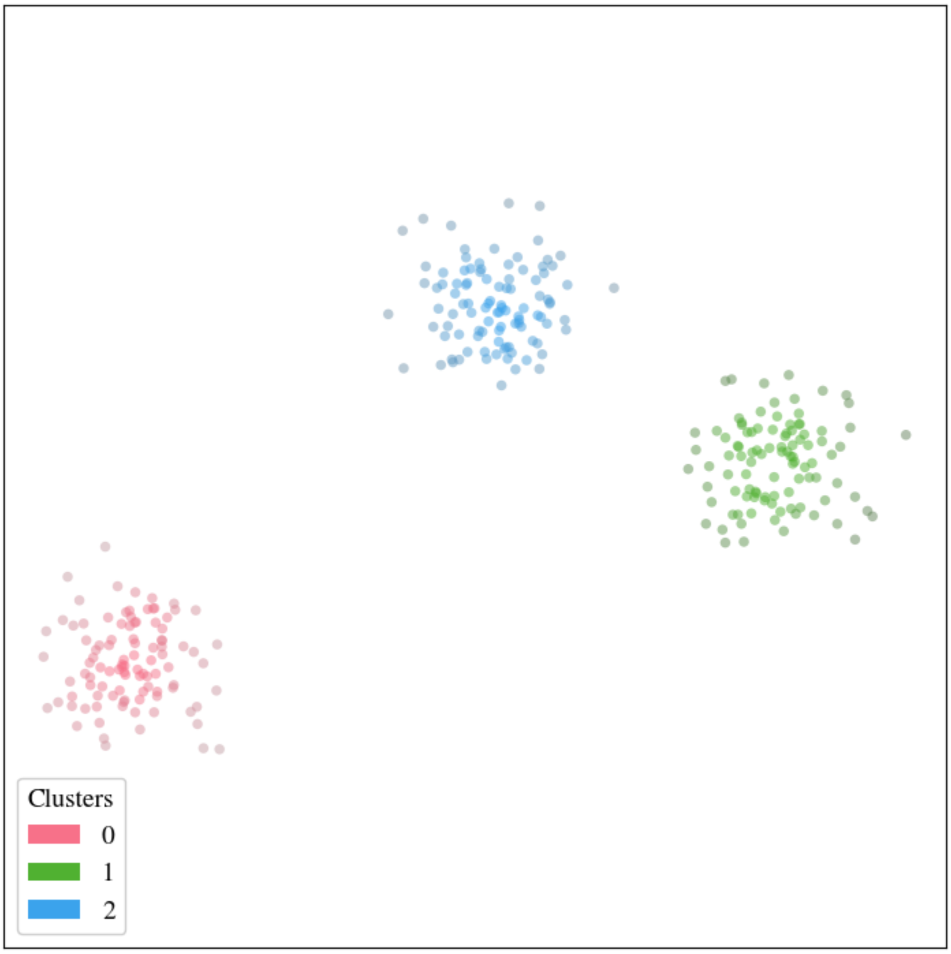


Figure 2: Pseudo-randomly generated points in a two dimensional space clustered into three distinct groups using HDBSCAN.

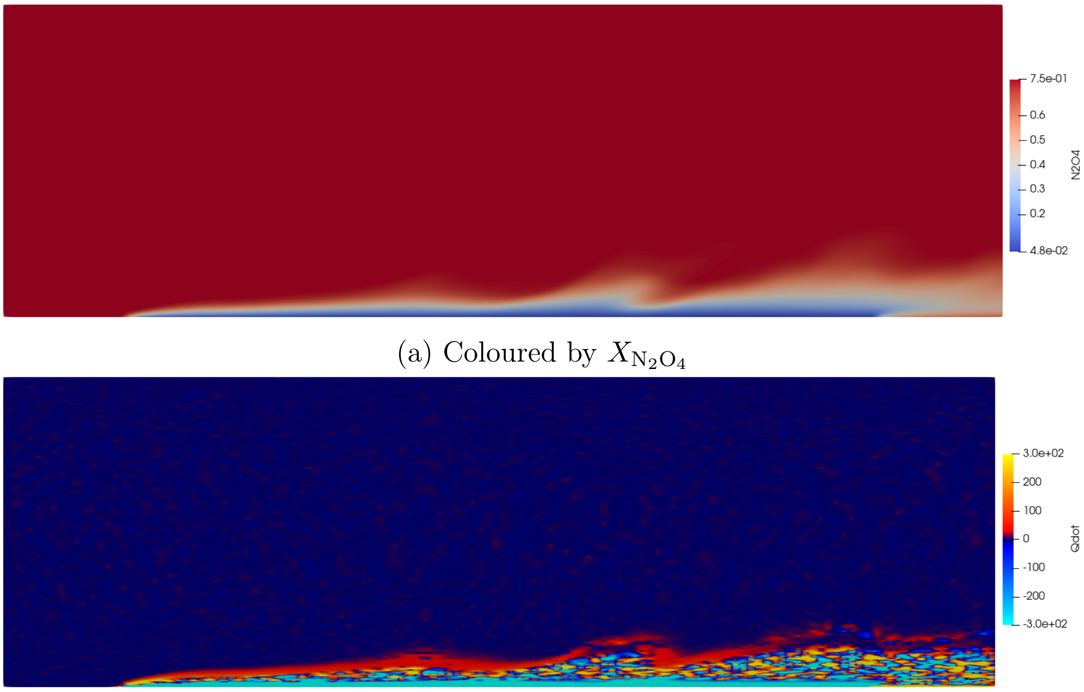
their interpretation varies with the exact algorithm and method employed. Examples of methods or algorithms for feature correlation include linear correlation and mutual information, the latter of which is used in the workflow of Rovira et al. [1].

## 1.5 Aim of Study

The workflow proposed by Rovira et al. [1] can potentially be used for gaining understanding of physical systems as well as explain them by classifying complex data into regions of distinct, simpler physics. It is presented as general and applicable to a wide range of problems in CFD, and potentially other fields.

In the present work, it has been applied to a more complex reactive flow data set from a study by Zhang et al. [2], to test its applicability to a wider range of data sets. The performance of the algorithms on this new data set is evaluated and compared to that shown in the work by Rovira et al. [1]. The consistency of the results over the time evolution of the system is also examined. The workflow has previously been applied only on data from a single time instance of a simulation.

The data set used comes from a simulation of $N_2O_4$ gas flowing through a channel with a hot section of the bottom wall. As it passes the hot wall, the gas decomposes in the reversible reaction $N_2O_4 \rightleftharpoons 2\,NO_2$, absorbing heat as chemical energy. Away from the wall, $NO_2$ reacts back into $N_2O_4$, releasing heat. How this process affects heat transfer away from the hot wall was investigated by Zhang et al. [2]. A possible application is in industrial cooling solutions. A snapshot from the simulation is shown in Figure 3.



(a) Coloured by $X_{N_2O_4}$

(b) Coloured by $\dot{Q}$

Figure 3: Snapshot 258/866 of the simulation coloured by $N_2O_4$ mass fraction (a) and $\dot{Q}$ (heat release) (b). Note the band of very low heat release (high heat absorption) along the hot section of the bottom wall.

# 2 Method

The data used in the present work consists of 866 time instances ("snapshots") taken 1.5 seconds apart, of a 2D slice of the original 3D simulation by Zhang et al. [2]. The data is described in more detail in Section 2.1. Each slice was first pre-processed as described in Section 2.2, and each step of the workflow was applied as described in Sections 2.3 to 2.5. The clusters were also mapped back onto the original mesh. The implementation of the workflow used in the present work[1] was done in Python and based on that of Rovira et al. [1][2], with some modifications[3] being made to the original source code for extended functionality and optimisation, without changes to the algorithms themselves.

## 2.1 Description of Data Set

The original simulation was a Direct Numerical Simulation, meaning that the governing equations, the Navier-Stokes equations, were numerically solved at every scale. This is in contrast to the paper by Rovira et al. [1], where the workflow is tested on data from an LES simulation, where turbulence modelling is used in place of directly solving the governing equations for the smallest scales. In the simulation, a gas mixture of 5% (by mass) inert $N_2$ and $N_2O_4$ and $NO_2$ at chemical equilibrium, that is with mass fractions of approximately 73% and 22%, respectively, enters the chamber at a temperature of 303 K (30.85 °C). A section of the bottom wall with a temperature of 404 K (130.85 °C) heats the gas, which causes the decomposition of $N_2O_4$ into $NO_2$, an endothermic reaction absorbing heat. As the gas mixes, $NO_2$ recombines back into $N_2O_4$ in areas with sufficiently low temperature, releasing heat. Each snapshot consisted of 70416 cells in a polygonal mesh, with 49409 points. As the snapshots were 2D slices, the cells were triangular. Unlike the original implementation of the workflow, the mesh has not been resampled onto a uniform one, and there was a higher number of cells close to the top

[1] Source code available at `https://github.com/Armadillan/rays`

[2] Original source code available at `https://github.com/marrov/keyfi`

[3] Modified source code available at `https://github.com/Armadillan/keyfi`

and bottom walls of the channel. Each point held information about the mass fractions of each chemical species, velocity along each axis, temperature, density, and heat release. The dimensions and symbols for these are presented in Table 1. For more details on the simulation setup, refer to the original work by Zhang et al. [2].

Table 1: Variables in the data set with units and symbols.

| Variable | Mass fraction | | | Velocity | | | Density | Temperature | Heat Release |
|---|---|---|---|---|---|---|---|---|---|
| | $N_2O_4$ | $NO_2$ | $N_2$ | $x$ | $y$ | $z$ | | | |
| Symbol | $X_{N_2O_4}$ | $X_{NO_2}$ | $X_{N_2}$ | $U_x$ | $U_y$ | $U_z$ | $\rho$ | $T$ | $\dot{Q}$ |
| Unit | dimensionless | | | $\mathrm{m\,s^{-1}}$ | | | $\mathrm{kg\,m^{-3}}$ | K | $\mathrm{W\,m^{-3}}$ |

The heat release variable, $\dot{Q}$, can be understood as the net energy released from chemical reactions within a cell per unit time, per unit volume of the cell. Because the value is normalised by the cell volume, values can be compared between cells in a non-uniform grid. If the reactions within a cell are primarily endothermic, where more energy is absorbed than released, $\dot{Q}$ is negative. It is defined by the equation

$$\dot{Q} = \sum_{k=1}^{N_{sp}} h_{f,k}^{0} \dot{\omega}_k, \tag{1}$$

where $N_{sp}$ is the number of chemical species, $\dot{\omega}_k$ is the rate of production of the $k^{\text{th}}$ species for every volume unit of a cell, which is negative if the species is consumed by chemical reactions, and $h_{f,k}$ is the enthalpy of formation for the $k^{\text{th}}$ species. In words, $\dot{Q}$ is the sum of the rates of production of each species multiplied by the energy released per unit mass of produced substance, that is, the net energy released into the environment, as heat, of all the chemical reactions happening in a given cell.

## 2.2 Pre-Pocessing

The input data was pre-processed by removing certain variables, clipping $\dot{Q}$, and scaling the variables. The variables removed were $U_z$, $X_{N_2}$, $X_{NO_2}$, and $\rho$. These variables were added back before the feature correlation step.

### 2.2.1 Dropping Variables

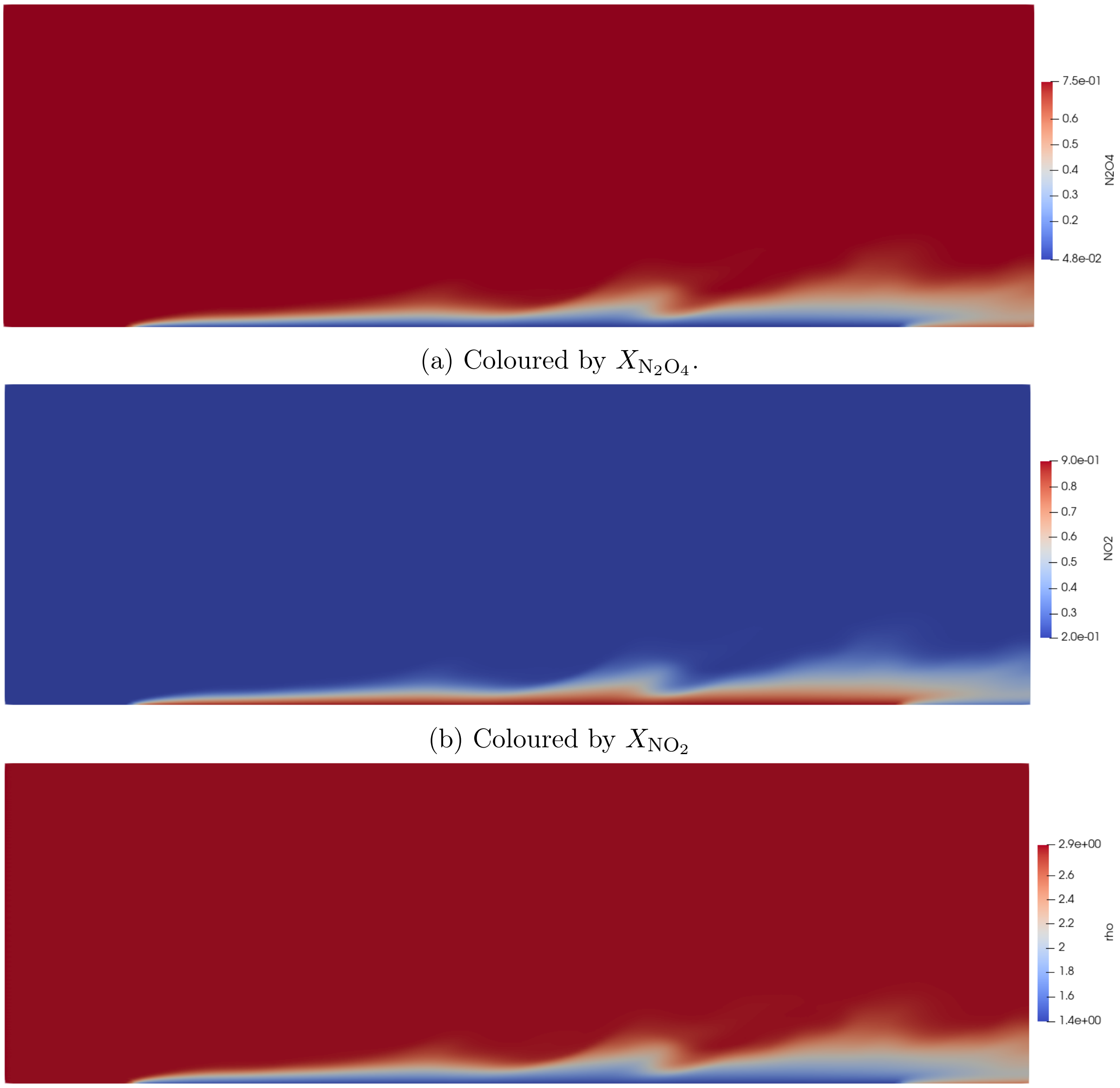
The rationale for the removal of $X_{\mathrm{N_2}}$ was that it is inert and does not participate in any chemical reactions in the fluid. It is not an interesting variable to analyse, and the variations in this variable are very small throughout all of the data. Including this variable would essentially be introducing noise, and could make it harder for the algorithms to find correlations between other variables.

The reason for not including $X_{\mathrm{NO_2}}$ and $\rho$ was that they have a direct and easily understood correlation with $X_{\mathrm{N_2O_4}}$. Because $X_{\mathrm{N_2O_4}}$ and $X_{\mathrm{NO_2}}$ are fractions of the total mass, and $X_{\mathrm{N_2}}$ stays practically constant, with a value of 0.05, they are related by $0.95 = X_{\mathrm{N_2O_4}} + X_{\mathrm{NO_2}}$. Pressure is practically constant throughout the simulation, and thus $\rho$ is dependent only on the mass fractions of $N_2O_4$ and $NO_2$, and the relative mass of these molecules. The only information added into the system by including $\rho$ would be the relative mass of $N_2O_4$ and $NO_2$, which should not have much influence on the chemical and physical processes simulated. Because all of these three variables hold essentially the same information, which is visualised in Figure 4, only one was included. The variable to be included was arbitrarily chosen to be $N_2O_4$.

The reason $U_z$ was dropped was that another dimension of velocity does not contribute a lot of useful information, given that the data is a 2D slice of the simulation. The workflow will not be able to identify 3D structures in the fluid flow anyway, because that information does not exist in 2D data. Including $U_z$ is therefore unlikely to reveal any patterns or relationships that the information contained in $U_x$ and $U_y$ would not. Dropping $U_z$ likely removes more noise than information from the data.

### 2.2.2 Clipping $\dot{Q}$

$\dot{Q}$ was clipped to the range $[-300, 300]$. Because the mesh is coarse the exact differences in this variable between cells with very high or very low heat release are not meaningful. Because the cells are so large, cells with very high heat release do not necessary correspond to points with very high heat release, but to general areas with high heat release, or



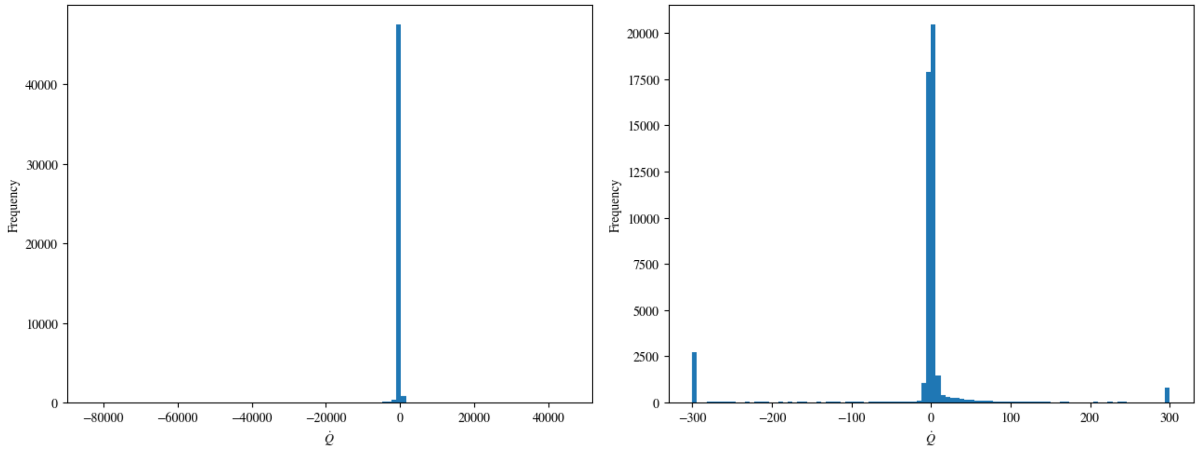
(a) Coloured by $X_{\mathrm{N_2O_4}}$.

(b) Coloured by $X_{\mathrm{NO_2}}$

(c) Coloured by $\rho$.

Figure 4: Snapshot 258/866 of the simulation coloured by $X_{\mathrm{N_2O_2}}$ (a), $X_{\mathrm{NO_2}}$ (b), and $\rho$ (c).

possibly areas with single points of very high heat release. Above and below a certain threshold, the heat release gradient between cells is not not of interest anymore, as a cell with very high heat release and one with maybe even many times that contain essentially the same information. A suitable range to clip $\dot{Q}$ was chosen by analysing the distribution of this variable. The distribution before and after clipping can be seen in Figure 5.



(a) Not clipped.

(b) Clipped to [-300, 300].

Figure 5: Histograms of $\dot{Q}$ values in snapshot 73/866 before (a) and after (b) clipping to [-300, 300].

The aim was that every cell above or below the range should contain the same information, that is "very high" or "very low" heat release, respectively, but that the distribution of heat release values above 0 remained clear and distinct from the data at the edges of the distribution, as this data could potentially hold interesting information about the boundaries between areas of very high and very low heat release, and the heat release gradient at the edges of the reacting region close to the bottom wall.

### 2.2.3 Scaling Data

The data was scaled by dividing by the maximum absolute value. The original workflow as implemented by Rovira et al. [1] scales the input data by subtracting the mean and dividing by the standard deviation. It was found that, for this particular data set, dividing by the absolute maximum value results in more distinct clusters. Temperature, heat release and $N_2O_4$ mass fraction were scaled independently, while $U_x$ and $U_y$ were scaled together, that is, each value of those two variables was divided by the maximum absolute value in both variables. This was done to preserve the relative magnitudes of the velocity components, and the overall velocity vector (as projected to 2D by dropping $U_z$). Scaling $U_x$ and $U_y$ separately gives too much significance to the much smaller $y$ component, the maximum value of which was an order of magnitude smaller than that of $U_x$, and whose mean was around three orders of magnitude smaller. When scaled separately, the dimensionality reduction using UMAP would produce hard to cluster “jellyfish" embeddings, an example of which can be seen in Figure 6.

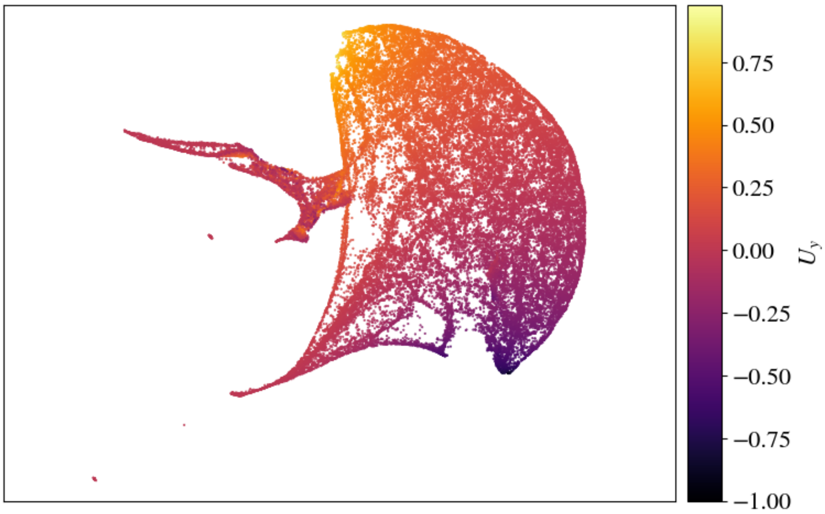


Figure 6: UMAP embedding of snapshot 73/866, with $U_x$ and $U_y$ scaled separately, coloured by $U_y$.

## 2.3 UMAP

After preparing the input data, each snapshot is passed to UMAP, for embedding the data into a 2D space. It should be noted that the physical coordinates of each point in the 3D space of the simulation are not included in the input data. The data at this point has 5 dimensions: temperature, heat release, mass fraction of $N_2O_4$, and velocity along the $x$ and $y$ axes.

UMAP is a non-linear neighbour-graph type dimensionality-reduction algorithm used based on fuzzy topology, developed by McInnes et al. [8]. It is not limited to any one type of data, and has, in addition to its use in CFD, been successfully applied in data mining and bioinformatics [9]. In broad terms, it works by creating a graph linking nearby points in the data set, and embedding this graph into a lower dimensional space, preserving information about the distances between points, the structure of the data. The work of Rovira et al. [1] gives a broader overview of this algorithm, and for an explanation of the mathematical backing of this algorithm, the reader is referred to the original paper by McInnes et al. [8].

Dimensionality reduction algorithms generally strive to preserve the global and local structure of the data. Local structure refers to the distances between each point and its closest neighbours, while global structure refers to the distances between each point and the points farthest from it. Preserving local structure can be understood as that dissimilar points in the original data set are not represented as similar in the embedding, while preserving global structure is in rough terms that similar points in the original data is not represented by dissimilar points in the embedding.

In contrast to simpler techniques like PCA or matrix factorisation, UMAP is not limited to capturing linear relationships in the data, and can potentially preserve any type of correlation of the original data in the embedding. It has also been shown to be better than other non-linear machine-learning algorithms like t-SNE, which was used in the work by Zhang et al. [2], at preserving the global structure of the original data. The dissimilarities between points in embeddings created by UMAP are generally more meaningful than with t-SNE, twhere dissimilar points in the embedding are more likely to be dissimilar in the higher-dimensional original data.

It should also be noted that, as mentioned in the introduction (Section 1.2), the coordinates of each point in the lower dimensional embedding, the “synthetic variables”, are not indicative of any of the properties of the original points. These values are assigned to each point by the algorithm in a way that preserves the distances between points, but not necessarily any other information about their position in the high-dimensional space, such as in which dimensions they are different.

The main hyperparameters in UMAP are the number of neighbours and the minimum distance between points. The number of neighbours dictates how many points will be considered for comparison with each point in the high-dimensional space. It controls the balance between preserving global and local structure. A high number of neighbours will generally retain more of the global structure in the embedding, while lower values will prioritise local structure. The minimum distance parameter defines the minimum distance between points in the embedding. Lower values will result in more tightly packed clusters. Rovira et al. [1] recommend setting the number of neighbours to $\sqrt{N}$ where $N$ is the number of points in the data set, and the minimum distance to 0.1. After trialling a range of hyperparameters on a subset of the data, values of 250 and 0.1 were chosen. These values generated distinct and relatively tight clusters that seemed to correspond to points with similar physical properties. All 866 runs of UMAP, on every snapshot, were initialised using the same random state seed, to minimise the influence of the stochastic components of the algorithm. An example embedding using these hyperparameters is shown in Figure 7.

## 2.4 HDBSCAN

The embeddings were then passed to HDBSCAN, which identified clusters of points in the lower-dimensional data. HDBSCAN, Hierarchical Density-Based Spatial Clustering of Applications with Noise, is a hierarchical clustering algorithm, meaning that the clusters identified are ordered in a hierarchical structure, with large clusters subdivided into smaller ones. HDBSCAN can therefore both cluster data into a smaller number of large clusters, or a higher number or small clusters. An example HDBSCAN clustering of a snapshot from the data set can be seen in Figure 8. HDBSCAN is able to classify data as noise, and does this by assigning it to a cluster with the label -1.

HDBSCAN was developed by Campello et al. [10]. The implementation used by Rovira et al. [1], and therefore in the present work, is that by McInnes and Healy [11], which has

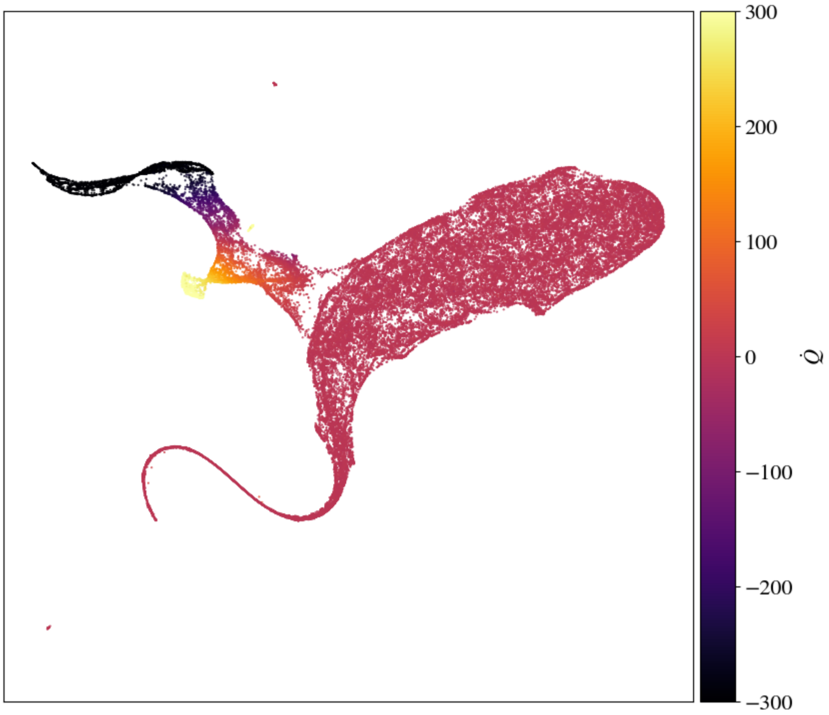


Figure 7: UMAP embedding of data points from snapshot 258/866, using 250 for the number of neighbours and 0.1 for minimum distance, coloured by $\dot{Q}$.

improved computational performance. For a more technical overview of the algorithm, and the mathematical motivations behind it, the reader is referred to these papers.

The main hyperparameters of HDBSCAN are the minimum cluster size and the minimum samples. Minimum cluster size is the minimum number of points that can be considered one cluster, while the minimum samples has to do with how HDBSCAN handles noise. Lower values will make the algorithm more "cautious", resulting in more data points being classified as noise and not included in any cluster. The values chosen for these parameters in the present study were 300 for minimum cluster size and 10 for minimum samples.

## 2.5 Mutual Information

So far in the workflow, clusters of similar points in the original data have been identified. The last step uses Mutual Information, for learning how the points in each cluster are similar. This algorithm is used to correlate the synthetic variables of points in each

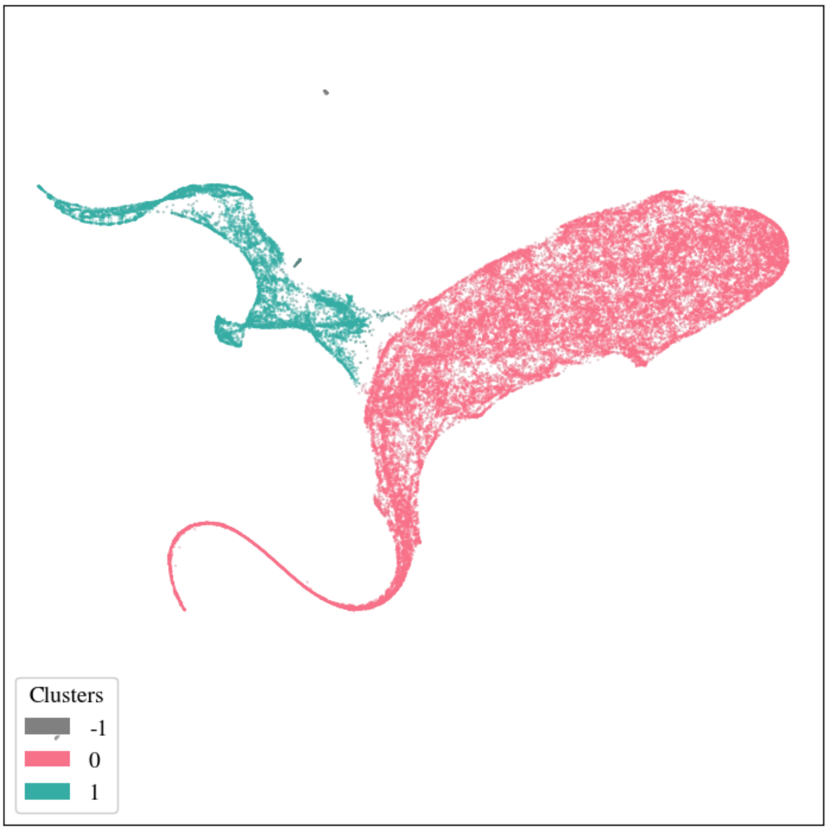


Figure 8: An HDBSCAN clustering of UMAP embedded data from snapshot 258/866, using a minimum cluster size of 300 and minimum samples of 10. Cluster -1 is noise as identified by HDBSCAN.

cluster to their variables in the original high-dimensional space. Here, all of the original variables are correlated with the synthetic variables, not only the five that were used for dimensionality reduction. Throughout this workflow, only non-linear methods were employed, meaning that any type of relationship between the variables of the data set should have been possible to preserve. It is therefore important that the algorithm used for the feature correlation step does not only find linear relationships, as it is known that other types of relationships can exist in the data. Mutual Information quantifies the amount of information learned about one variable by studying another, and can therefore be used to measure any type of dependency between two variables.

The concept of Mutual Information was first introduced by Shannon in *A Mathematical Theory of Communication* [12], the founding work of the field of information theory. It became widely used as a way to describe features of and correlations within a data set after a paper by Battiti [13].

## 2.6 Mapping the clusters back onto the original mesh

It can be interesting to see where the clusters of points identified by the workflow exist in the two-dimensional physical space of the original slice of the simulation (for each cluster). To this end, the original mesh was updated with labels for each point based on which cluster it was placed in by HDBSCAN. A snapshot coloured by cluster is shown in Figure 9.

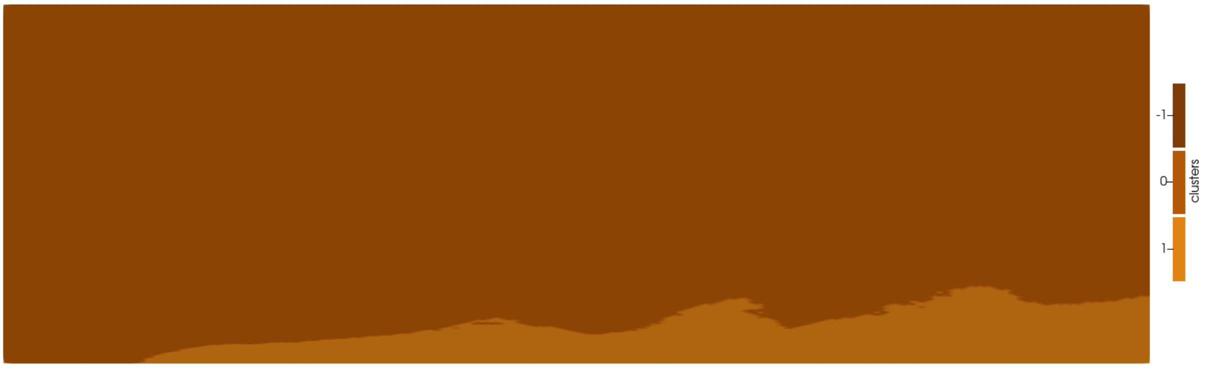


Figure 9: Embedding of snapshot 258/866 coloured by HDBSCAN clusters. Cluster -1 is noise as identified by HDBSCAN.

# 3 Results

Overall, the steps of the workflow successfully identified regions in the data with similar physical characteristics. In a majority of snapshots, two main clusters were identified in the embedding, like in the example in Figure 10. Mapping these clusters back onto the original data, as seen in Figure 11, it can be seen that the two main identified areas are the main flow and the turbulent area close to the hot bottom wall.

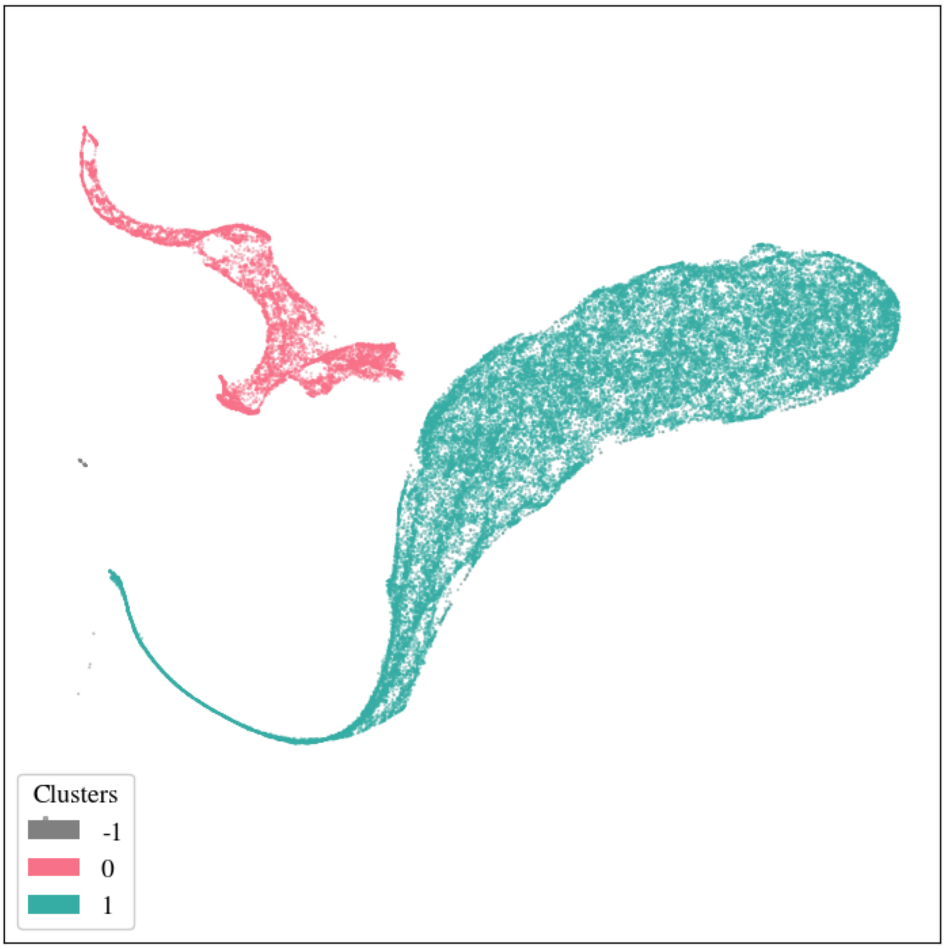


Figure 10: UMAP embedding of snapshot 22/866, coloured by clusters found by HDBSCAN. Cluster $-1$ is noise as identified by HDBSCAN. The axes have no labels, as the coordinates of each point in the embedding, the “synthetic variables”, hold no physical meaning.

The Mutual Information scores, which are shown for clusters 0 and 1 in this snapshot in Figure 12, suggest physical properties that can be expected of these regions. Cluster 0 shows a lot of variation in $\dot{Q}$, $\rho$, $X_{\mathrm{N_2O_4}}$, and $X_{\mathrm{NO_2}}$, and some variation in velocity,

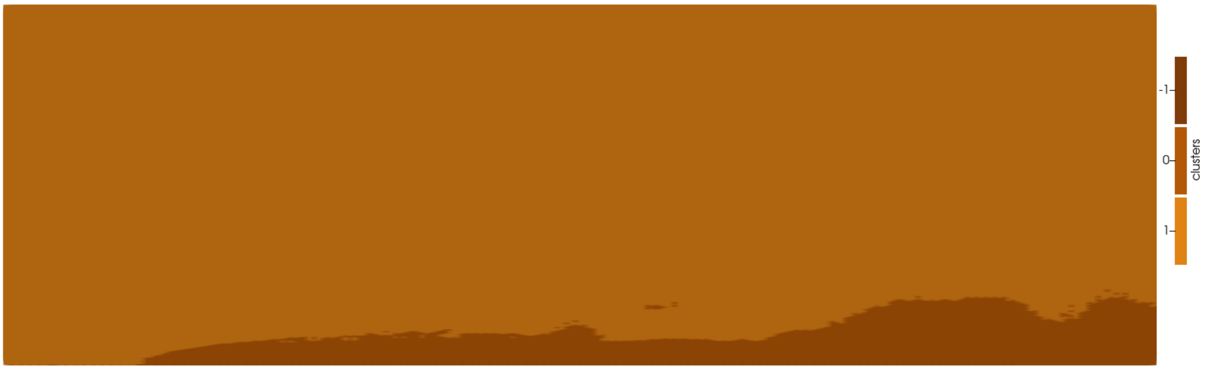
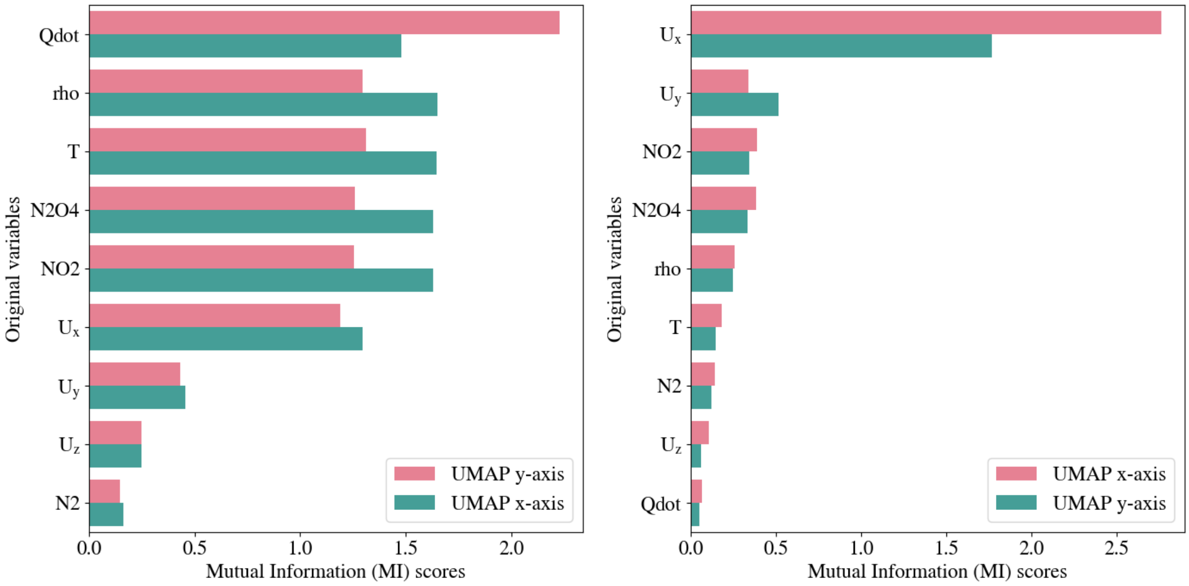


Figure 11: Clusters identified in the embedding of snapshot 22/866, mapped back onto the original data. Cluster $-1$ is noise as identified by HDBSCAN.



(a) Cluster 0.

(b) Cluster 1.

Figure 12: MI scores for clusters 0 (a) and 1 (b) in snapshot 22/866. Shown are MI scores between each variable in the data set and each of the two synthetic variables (UMAP x- and y-axes).

particularly along the x-axis. This suggest that many chemical reactions are happening there, because of the large variation in chemical species, density, and $\dot{Q}$. It also suggest some mixing and turbulence, given the variations in velocity. In contrast, cluster 1 shows little variation in variables other than $U_x$, which suggests that not a lot of chemical processes are taking place here. The large variation in $U_x$ is likely due to differences between slower moving fluid along the walls and the main flow in the middle of the channel. This will be discussed further in Section 3.3.

## 3.1 Finer Detail in Reacting Region

In some snapshots, some finer detail was identified in the reacting region, as seen in Figure 13. The clusters here correspond primarily to areas of very high and very low heat release, which can be seen in Figure 14. The MI scores for a selection of clusters in this snapshot are shown in 15.

## 3.2 Cases with Many Small Clusters

In a small number of snapshots, HDBSCAN divided the embedding into around 30 clusters. An example of this is shown in Figure 16.

## 3.3 Nearly-Distinct Clusters: Manual Labelling

Some embeddings looked as if the main flow cluster was splitting into multiple smaller clusters, like the one shown in Figure 17. A dendrogram showing the hierarchical relationship between possible clustering of this embedding is shown in Figure 18. These areas were manually marked as separate clusters, shown in Figure 19.

When mapped back onto the original mesh, shown in Figure 20, the clusters seem to correspond to areas of slower, more turbulent flow along the walls, and areas of high velocity in the main flow.

The MI Scores for the manually selected clusters are shown in Figure 21. Some em-

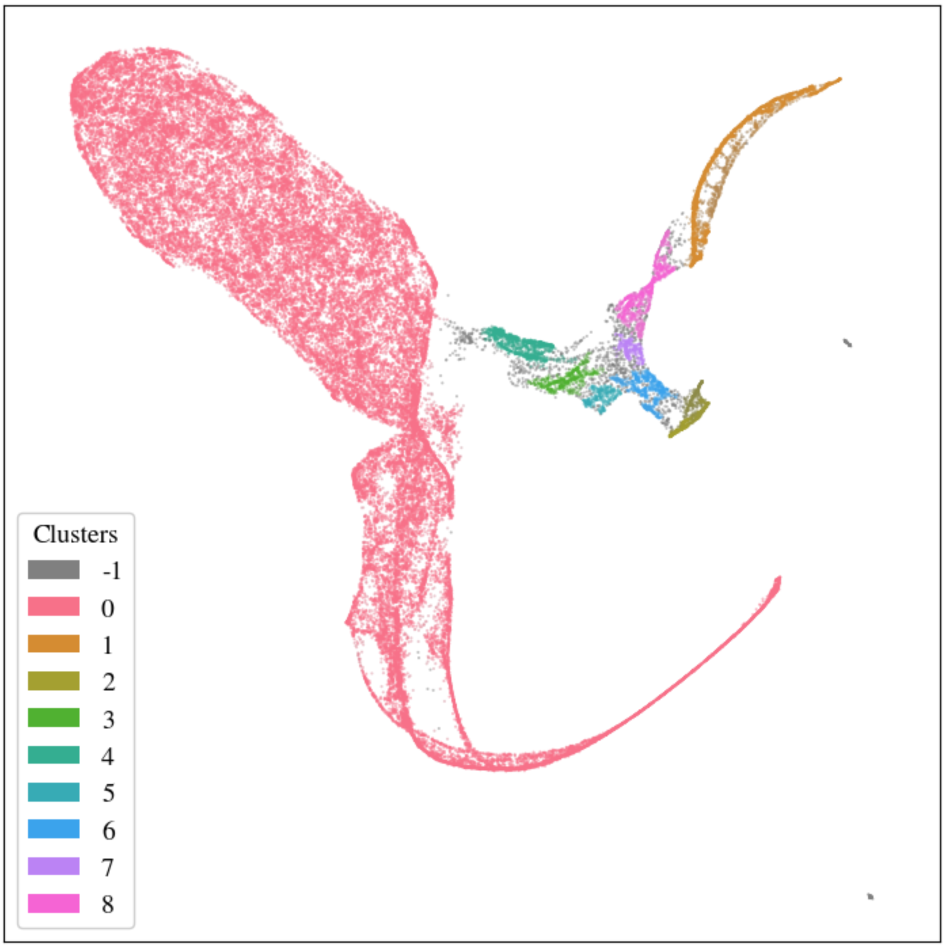
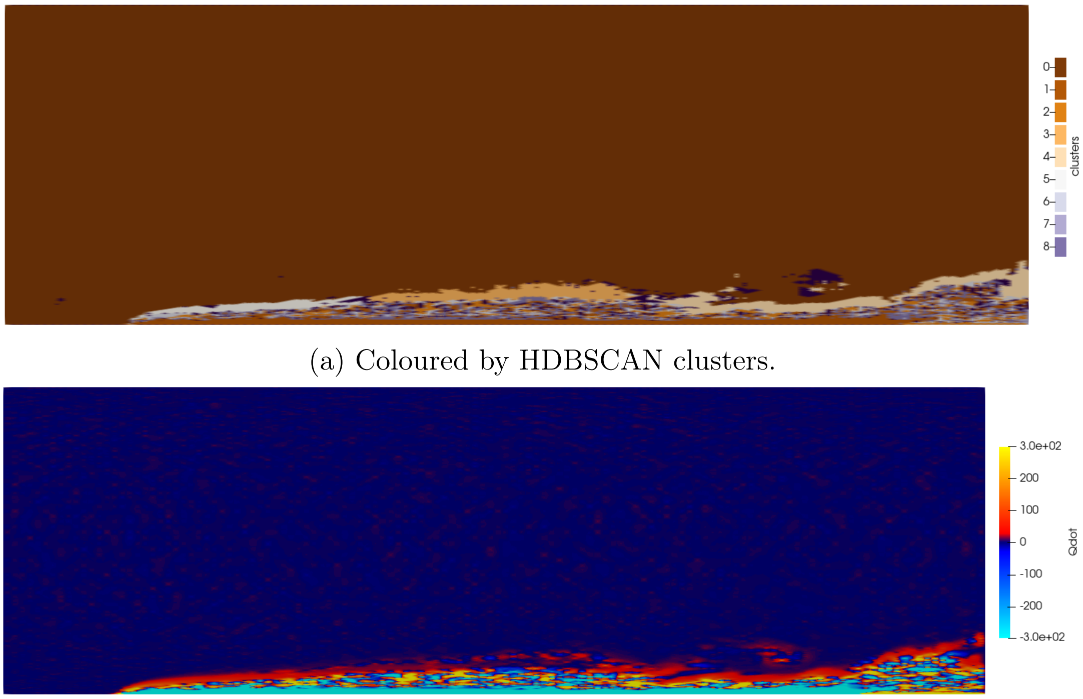


Figure 13: UMAP embedding of snapshot 84/866, coloured by clusters found by HDBSCAN. Cluster $-1$ is noise as identified by HDBSCAN. The axes have no labels, as the coordinates of each point in the embedding, the “synthetic variables“, hold no physical meaning.



(a) Coloured by HDBSCAN clusters.

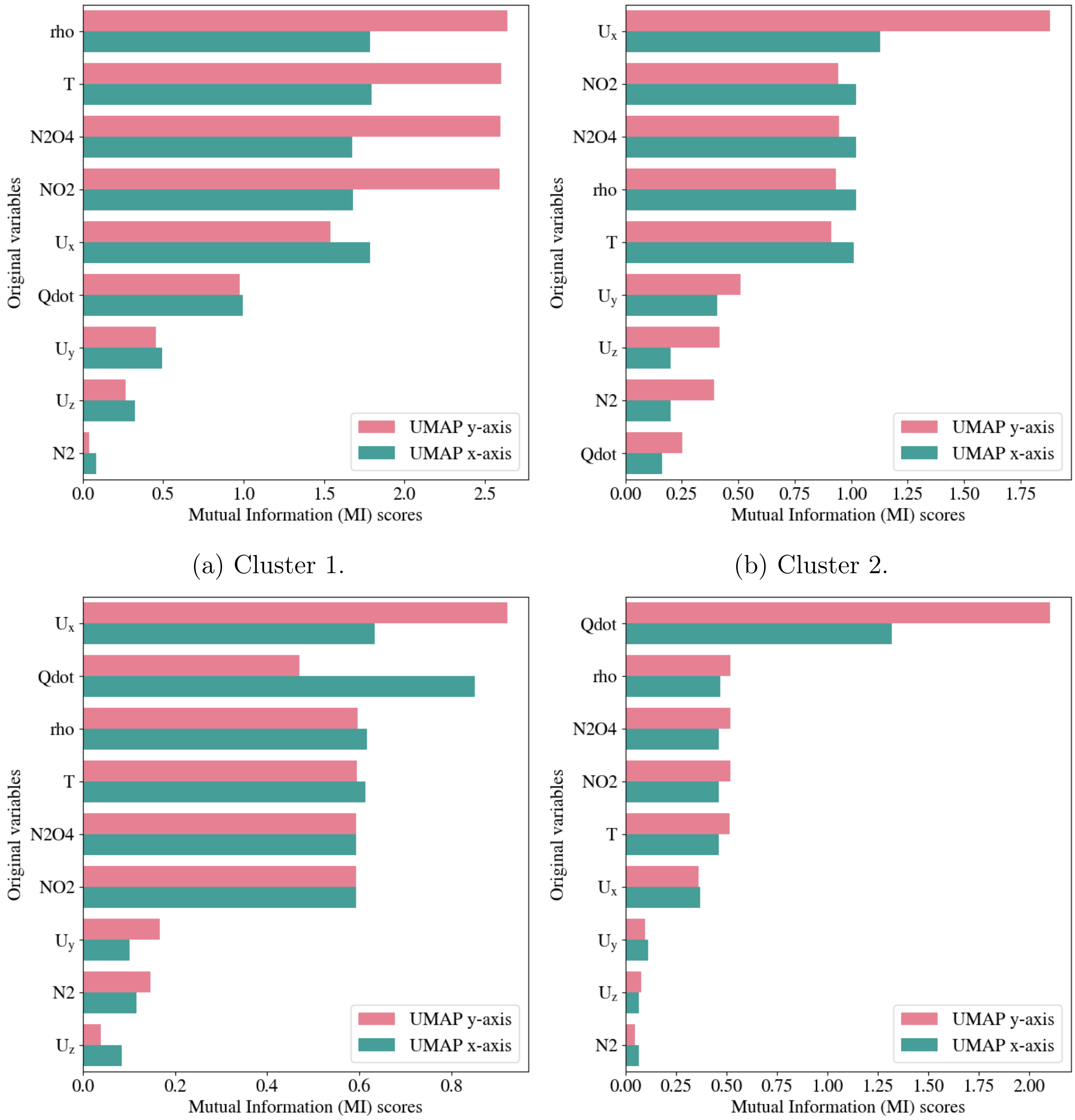
(b) Coloured by $\dot{Q}$

Figure 14: Snapshot 84/866 of the simulation coloured by HDBSCAN clusters (a) and $\dot{Q}$ (b).

beddings exhibiting this phenomenon also showed more detail in the reacting region, like the one shown in Figure 22.

## 3.4 Performance over time

The results are relatively consistent over time, identifying areas of roughly the same physical properties in most snapshots. The occurrence of each type of result described above is somewhat evenly distributed among the 866 snapshots. The clustering follows the time evolution of the system well.



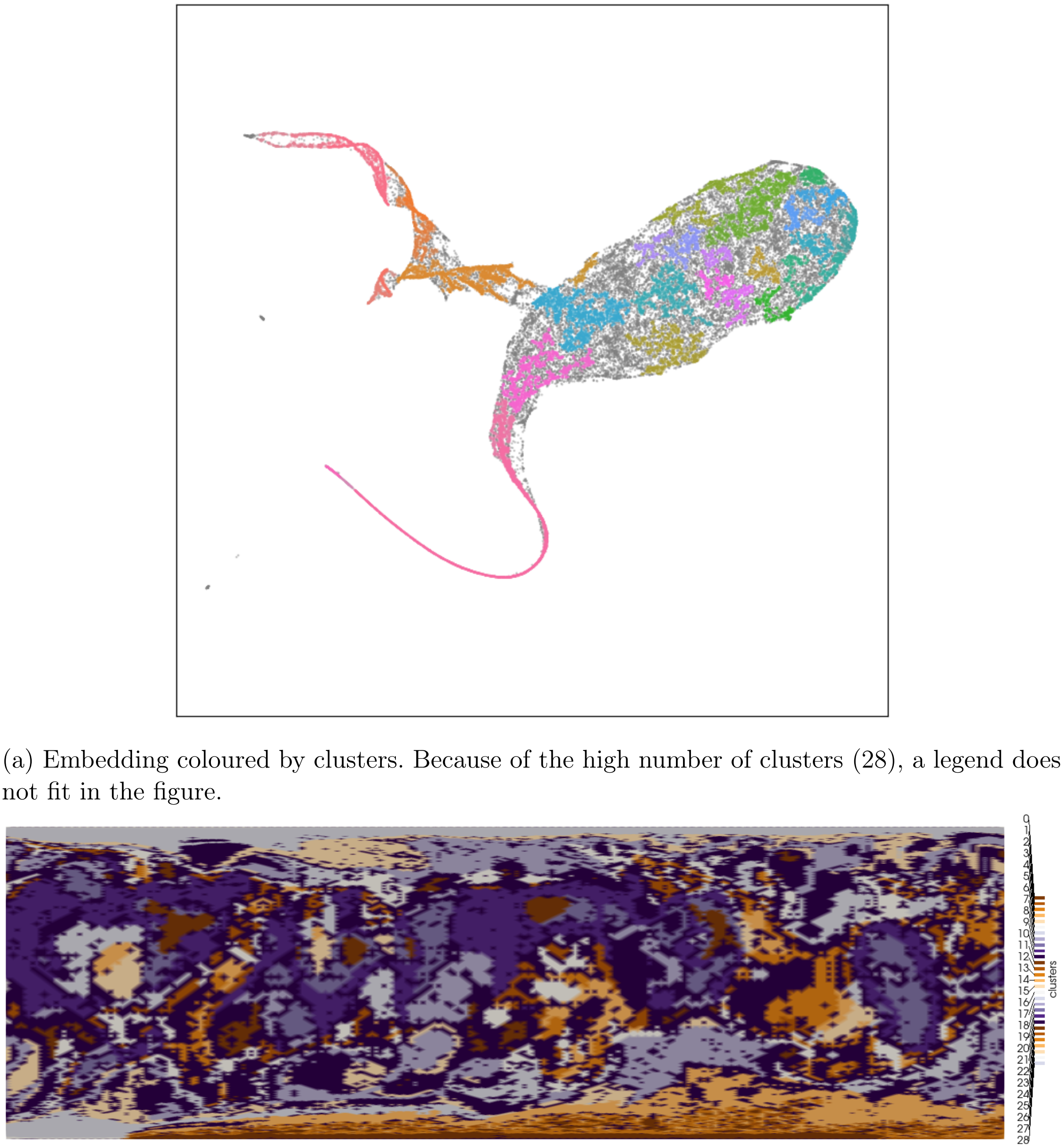
(a) Cluster 1.

(b) Cluster 2.

(c) Cluster 4.

(d) Cluster 8.

Figure 15: MI scores for clusters 1 (a), 2 (b), 4 (c), and 8 (d) in snapshot 84/866. Shown are MI scores between each variable in the data set and each of the two synthetic variables (UMAP x- and y-axes).



(a) Embedding coloured by clusters. Because of the high number of clusters (28), a legend does not fit in the figure.

(b) Snapshot coloured by clusters.

Figure 16: Clusters identified by HDBSCAN in the UMAP embedding of snapshot 556/866 (a), and mapped onto the original mesh (b).

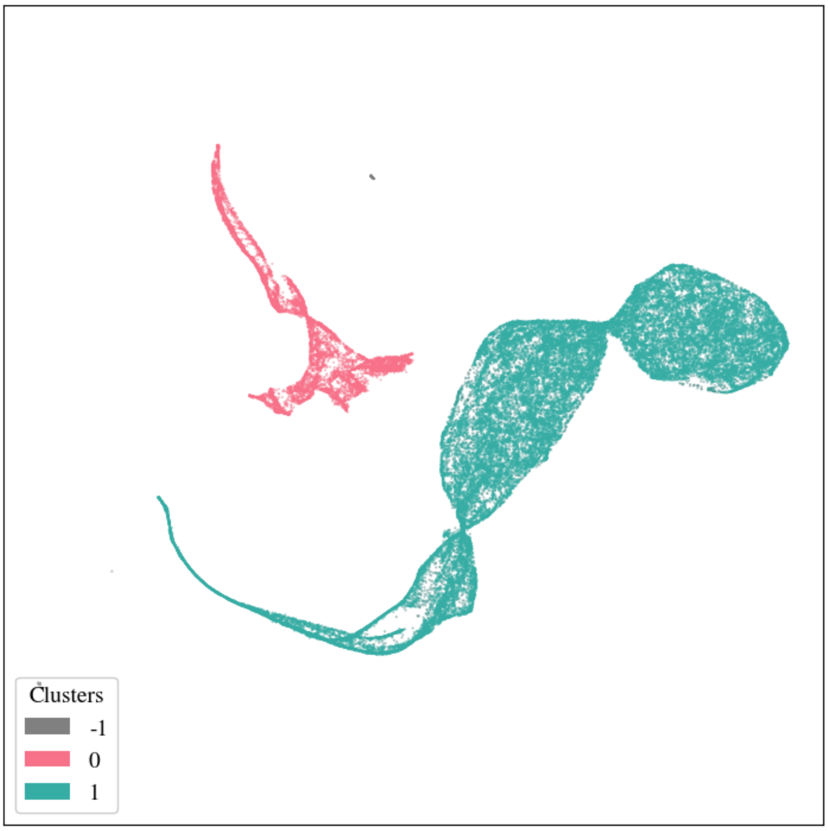


Figure 17: Clusters identified in the embedding of snapshot 73/866.

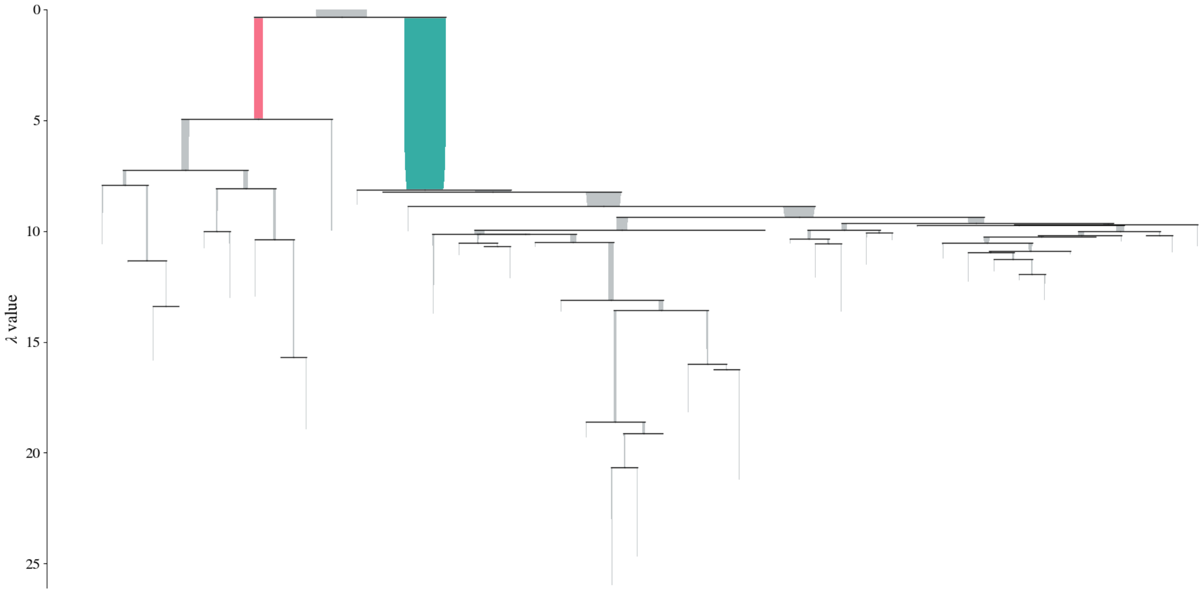


Figure 18: Dendrogram showing the hierarchical relationships betwen possible clusterings of data from snapshot 73/866. Clusters 0 and 1 as chosen by HDBSCAN are highlighted.

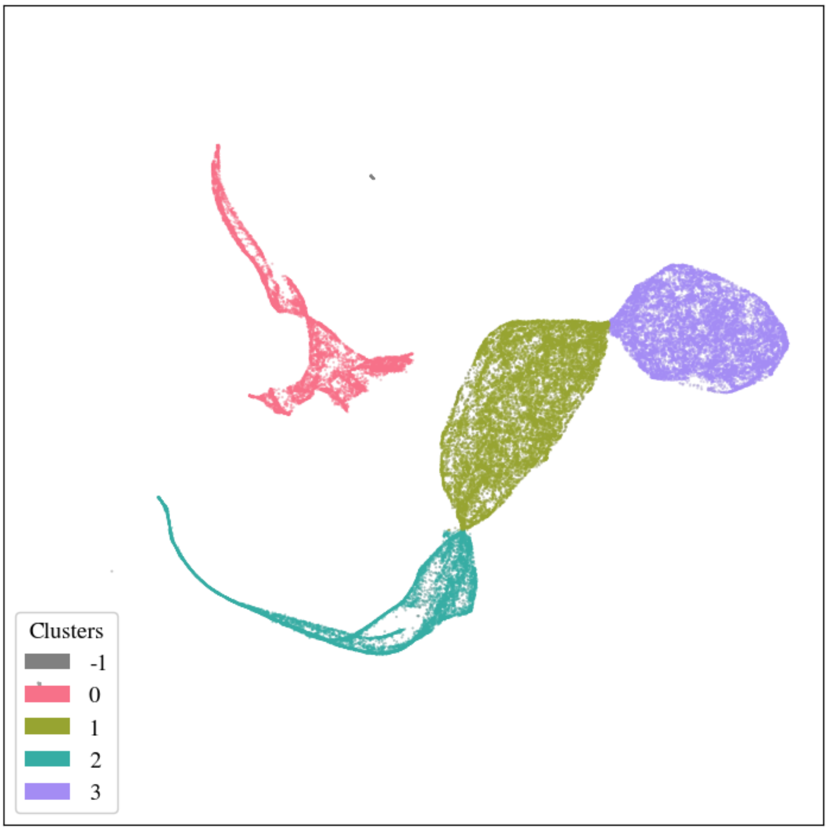
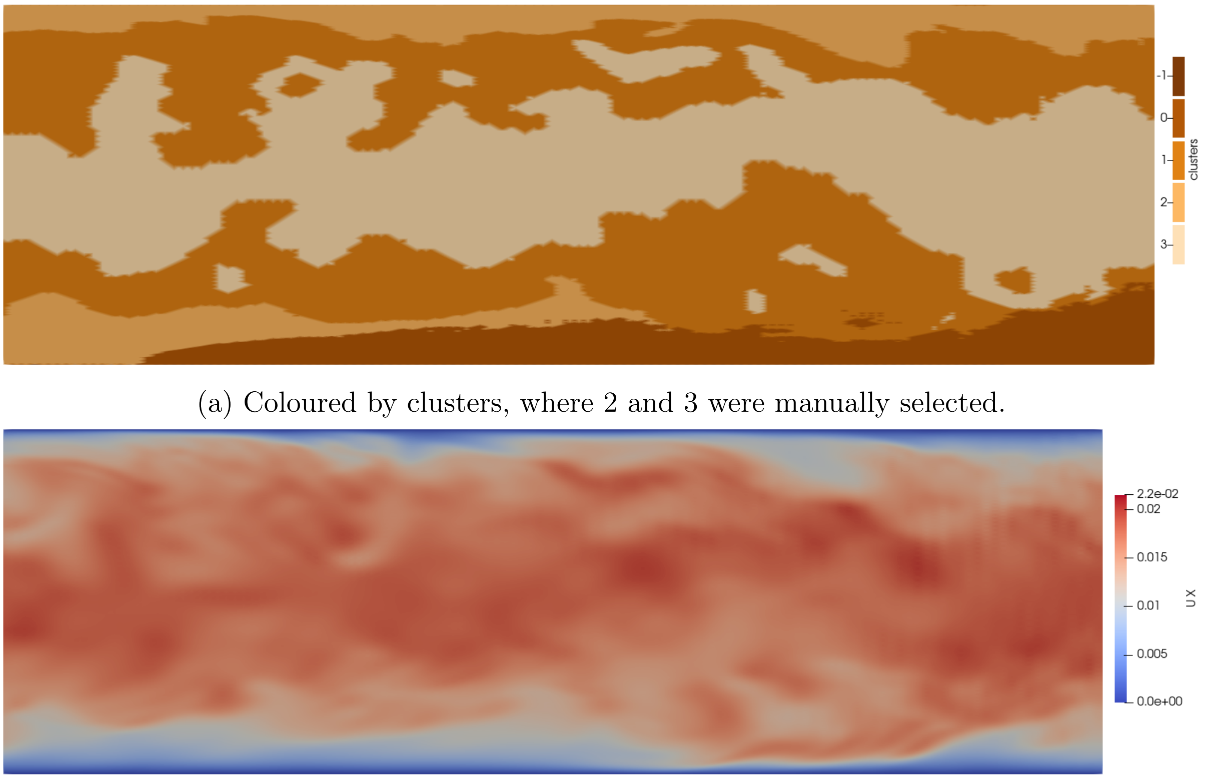


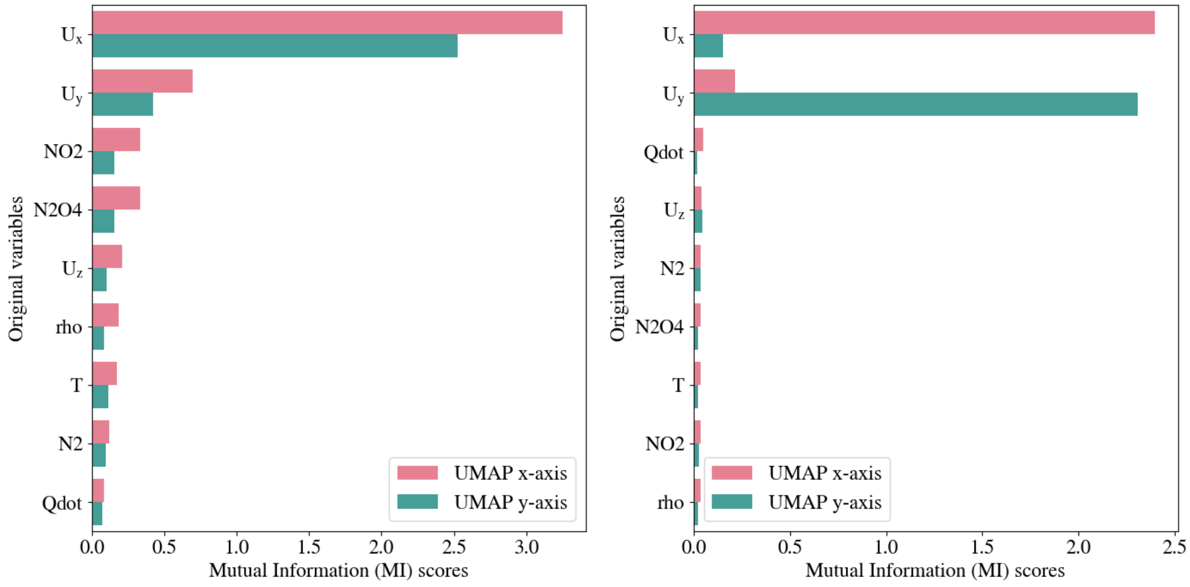
Figure 19: The embedding of snapshot 73/866, with clusters 2 and 3 manually selected.



(a) Coloured by clusters, where 2 and 3 were manually selected.

(b) Coloured by $U_x$.

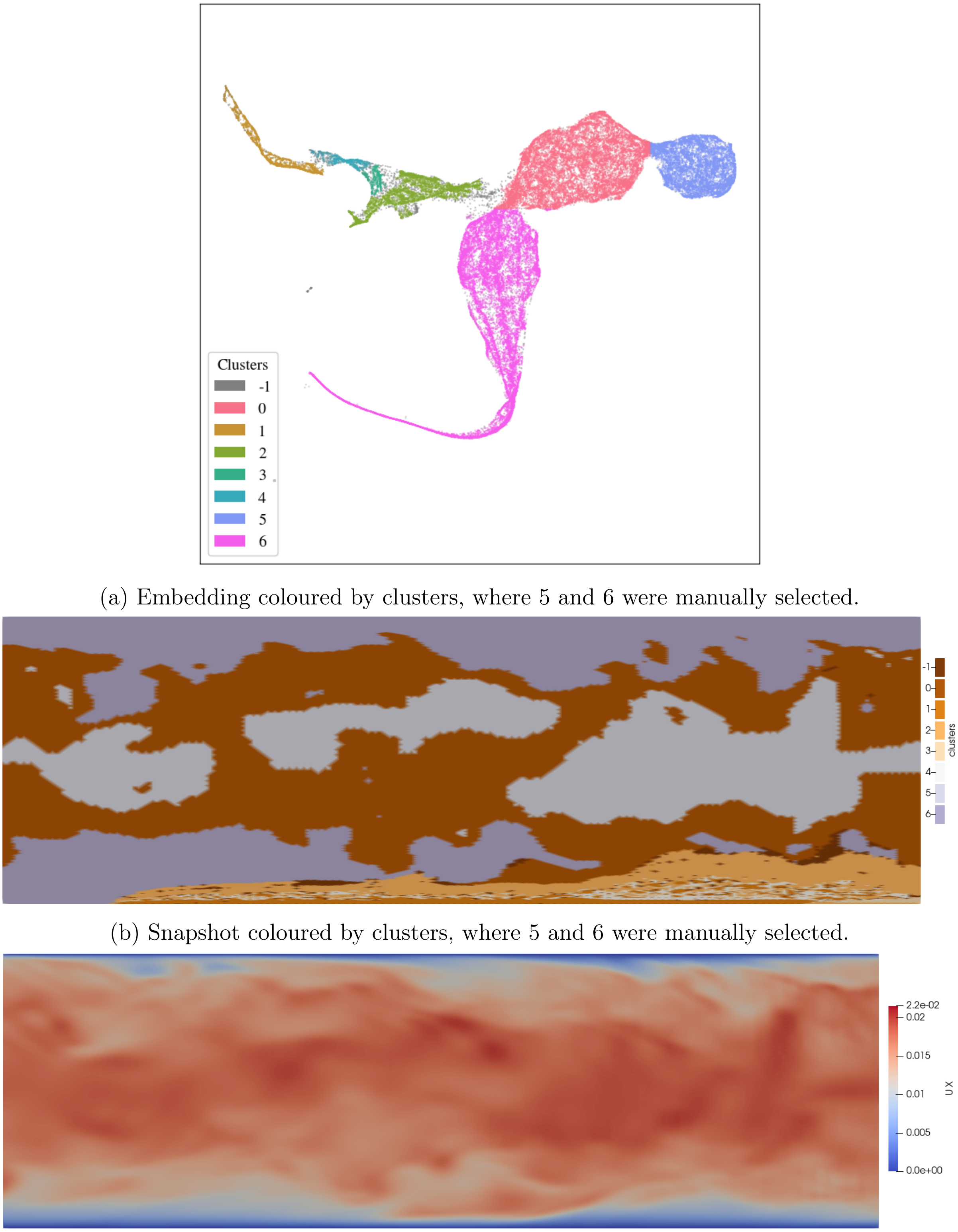
Figure 20: The manually selected clusters 2 and 3 mapped onto the original mesh of snapshot 73/866, together with the rest of the labels created by HDBSCAN (a), and the same snapshot coloured by $U_x$ (b). Clusters 2 and 3 were part of cluster 1 in the original HDBSCAN clustering.



(a) Cluster 2.

(b) Cluster 3.

Figure 21: MI scores for the manually selected clusters 2 (a) and 3 (b) in snapshot 22/866.



(a) Embedding coloured by clusters, where 5 and 6 were manually selected.

(b) Snapshot coloured by clusters, where 5 and 6 were manually selected.

(c) Snapshot coloured by $U_x$.

Figure 22: The embedding of snapshot 9/866, with manually selected clusters 5 and 6, coloured by cluster (a), mapped onto the original mesh, together with the rest of the labels created by HDBSCAN (b), and the same snapshot coloured by $U_x$ (c). Clusters 5 and 6 were part of cluster 0 in the original HDBSCAN clustering.

# 4 Discussion

The workflow proposed by Rovira et al. [1], as implemented in the present work, has successfully identified regions of similar physical properties and processes. It was relatively simple to understand, implement and adapt. The results obtained suggest that the workflow is useful for automatically detecting boundaries between regions in the simulation, identifying similar points at both large and small scales, and describing the defining features of each identified region. In essence, the workflow can be used to reduce the complex problem of understanding relationships between variables in a high-dimensional data set to several simpler problems, by reducing the data to regions of simpler physics. Additionally, the same general regions in the data, a region of main flow and a reacting region, were identified over most of the time evolution of the simulation. The phenomenon of clusters visually splitting into three parts corresponding to regions of different physical properties in the simulation, explained in Section 3.3 and Figures 17 to 22, is likely related to the low number of cells in the data set used in the present work. As a density based clustering algorithm, HDBSCAN needs enough data to identify and separate dense areas into different clusters. [10]. If there is not enough data the clusters will not be dense enough to be clearly separable. It is possible that with a finer mesh the data set would have enough points for these areas to be dense enough to be recognised as separate clusters by HDBSCAN. A greater number of points might yield better results with this workflow when applied on any data set, but comes at the cost of greater compute time. The balance likely has to be determined for each use case.

## 4.1 Possible Applications and Future Research

In addition to those mentioned in the original paper, several possible applications of this workflow or potential future research directions can be highlighted.

### 4.1.1 Consecutive Runs of HDBSCAN

In the case of the data set used in the present study, it is clear that both regularly identified “main” clusters, the main flow and reacting regions, can be further divided. The reacting region has in some snapshots been subdivided into regions such as the very low heat release and very high temperature region along the hot section of the wall, the transitory “gradient” between the reacting region and the main flow, and many areas of very high and very low heat release. The main flow could potentially be subdivided into slower moving, more turbulent regions close to the walls and a fast moving main flow in the middle of the channel. To investigate further, more fine divisions of the data set, or potentially other data sets this workflow is applied to, HDBSCAN can be re-run on just the points of single clusters identified in the original run. In other words, the initial clustering can be used as a mask for consecutive runs through HDBSCAN or other parts of this workflow, for a powerful and automated way of finding and describing fine structure in the data.

### 4.1.2 Model Order Reduction

Model order reduction is an area of research common in CFD. Reducing a complex model to a simpler one yielding the same or very similar results is beneficial, primarily in reducing computational complexity and improving explainability [6]. The robustness of this workflow, and consistency of results over time, suggests that examining the relationships between embeddings of consecutive snapshots could reveal relationships between the underlying data. It is possible that the output of this workflow could be used for predicting the evolution of the data, at least in terms of the properties of clusters in the data, and possibly the movement of points between them. This would reduce the reliance on computationally expensive simulations.

A problem to be solved before investigating this use case is how to automatically label similar regions the same over multiple snapshots. That is, how to ensure that for example the main flow region uses the same label over every snapshot. This should be possible to solve using the MI scores for each cluster, which describe the clusters well and relatively uniquely. A possible method is that a data set of clusters identified throughout many snapshots with the features being their MI scores is first constructed. Then a slightly modified version of this workflow could be applied on this data set, and identify categories of similar clusters, which would thus be automatically labelled.

### 4.1.3 Analysis of Hierarchical Relationships Between Data

Constructing hierarchical relationships between data is the foundation of HDBSCAN. In cases where a bigger cluster can be split into smaller ones, as in the cases shown in the present work, studying the hierarchical relationships between clusters could lead to greater insight into the structure of the underlying data. Generally, analysing the tree structure of possible clusters produced by HDBSCAN, like the one visualised by a dendrogram in Figure 18, could lead to greater understanding of the relationships between the various physical and chemical processes involved. This could be an alternative to further clustering the data, either by hand or algorithmically. This likely extends to other types of data sets, which can be categorised, and the categories subdivided, for a more fundamental model of the data.

### 4.1.4 Applicability to Other Fields

As suggested in the original paper, no component of this workflow is unique to CFD. It should in theory be applicable to any type of high-dimensional data set. The ease of use, useful results, and robustness of this workflow warrants consideration for use in other fields. It can at the very least be used for initial exploration of a data set, due to the ease and speed with which a general clustering of a high-dimensional data set can be achieved, together with relatively easily interpreted and useful description of the key features of each cluster.

## 4.2 Conclusion

In conclusion, the workflow developed by Rovira et al. [1] has in the present study been successfully applied to a different complex reactive flow data set. It automatically identified and described regions of the simulation with distinct physical and chemical properties and processes. It was found to be consistent, robust, and widely applicable, as well as relatively simple to understand and apply and adapt to a particular case. It was able to find both global, over-arching structure in the data, as well as more detailed local structure. Certain limitations were found stemming from the nature of the algorithms used, such as the reliance on big enough quantities of data, which in CFD is only a question of compute time. On the other hand, the workflow can potentially lower the computational power required for data analysis, due to its efficiency and effectiveness, and the potential use case of model order reduction. The workflow was found to be versatile and, seems adaptable to many use cases, from initial analysis to discovering fundamental structure in data. New possible applications and further research directions include analysis of hierarchical relationships between data, model order reduction, and applying re-applying parts of the workflow using the identified clusters as a mask, to find more complex, finer, or fundamental relationships in data, as well as the potential applicability of this workflow or an adaptation of it to fields outside CFD.

# References

[1] Marc Rovira, Klas Engvall, and Christophe Duwig. Identifying key features in reactive flows: A tutorial on combining dimensionality reduction, unsupervised clustering, and feature correlation. *Chemical Engineering Journal*, 438:135250, 2022. ISSN 1385-8947. doi: https://doi.org/10.1016/j.cej.2022.135250. URL `https://www.sciencedirect.com/science/article/pii/S1385894722007549`.

[2] Kai Zhang, Yazhou Shen, and Christophe Duwig. Identification of heat transfer intensification mechanism by reversible n2o4 decomposition using direct numerical simulation. *International Journal of Heat and Mass Transfer*, 182:121946, 2022. ISSN 0017-9310. doi: https://doi.org/10.1016/j.ijheatmasstransfer.2021.121946. URL `https://www.sciencedirect.com/science/article/pii/S0017931021010516`.

[3] Xuan Huang, Lei Wu, and Yinsong Ye. A review on dimensionality reduction techniques. *International Journal of Pattern Recognition and Artificial Intelligence*, 33 (10):1950017, 2019. doi: 10.1142/s0218001419500174.

[4] Yunus Cengel and John Cimbala. *Fluid Mechanics Fundamentals and Applications.* McGraw Hill, 2013.

[5] Ehsan Fooladgar and Christophe Duwig. A new post-processing technique for analyzing high-dimensional combustion data. *Combustion and Flame*, 191:226–238, 2018. doi: 10.1016/j.combustflame.2018.01.014.

[6] Toni Lassila, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Model order reduction in fluid dynamics: Challenges and perspectives. *Reduced Order Methods for modeling and computational reduction*, pages 235–273, 2014.

[7] Ira Assent. Clustering high dimensional data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):340–350, 2012.

[8] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection for dimension reduction. *Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861.

[9] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W H Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature Biotechnology*, 37(1): 38–44, 2018. doi: 10.1038/nbt.4314.

[10] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[11] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE, 2017.

[12] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[13] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550, 1994. doi: 10.1109/72.298224.