Technische Universität München
Fakultät für Informatik
Lehrstuhl für Computer Grafik und Visualisierung

Data-driven Ensemble Visualization

Alexander Kumpf

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr. Michael Georg Bader
Prüfer der Dissertation:	1. Prof. Dr. Rüdiger Westermann
	2. Prof. Dr.-Ing. Lars Linsen

Die Dissertation wurde am 05.03.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 22.06.2020 angenommen.

To my family and friends

Abstract

Numerical simulations have an inherent sensitivity to input parameters, which in many cases are not estimated perfectly. By using different parameterizations, an ensemble of simulations is generated in order to sample from the space of possible outcomes. Investigating every single member of this ensemble separately is tedious and often not possible, which is why major trends and similarities in the ensemble members are searched, introducing additional uncertainty by the methods used therefore. Further, dependencies between different quantities at different forecast times can reveal which errors propagate in time and where additional measurements might have improved the accuracy of the simulation. When relating differences in the realizations to initial parameters, metrics have to be found to compare different simulation outcomes.

Clustering analysis is used to group similar members together; however, its outcome varies depending on the specific parameters used for the clustering. By sampling systematically in the clustering space, we create an ensemble of clusterings hinting at sensitivities present. Multiple linked views support the user in finding a suitable clustering and provide insights into the quality and effects of it. Summary visualizations show the variability of cluster sizes, cluster-memberships of each data-point as well as uncertainties in feature visualizations such as contour plots. Specific clusterings or clusters can be selected and all views are updated interactively. Using these methods, we present a workflow assisting the user in clustering tasks which is applied exemplary on the forecasts of tropical cyclone Karl in 2016.

Correlation analysis in the ensemble data gives hints on dependencies between different regions, quantities and time steps. Appearing correlation patterns are sensitive to the exact locations correlated to each other, and correspondences between patterns over time-steps have to be established. We present a grid point clustering algorithm which provides a statistically coherent region to which other quantities are correlated to. By further clustering spatial structures of high correlation, regions of coherent features in the ensemble are indicated. Connected correlation structures are automatically tracked over time by combining optical-flow-based forward and backward tracking. The tracking quality and confidence is controlled using a split-merge diagram and a color coded swipe-path gives an immediate overview over a selected tracked structure. In the above mentioned real case, we were able to relate uncertainties in the exact position of cyclone Karl with the erroneous forecast of a heavy precipitation event in Norway a few days later.

The impact of changes in initial simulation parameters is important to know especially if the correct parameterization is not known. When simulations of objects with different size and location such as growing clouds are compared, correspondences between all simulation elements over different ensemble members are impossible to find. Instead, the multi-parameter distributions of simulation elements can be explored. Differences in constructed cumulative distribution functions (CDF) for each set of simulation elements serve as a comparison measure between them. The combination of multiple k-Means clusterings of t-SNE projected simulation elements via evidence accumulation finds clusters of arbitrary shape which can be compared over the ensemble. Clusters are matched using the same CDF-based measure and weighted accumulated cluster distances quantify the similarity between the sets. Parallel coordinates are then used to investigate the temporal evolution of matched clusters and their variability over the ensemble. Ultimately, distance are set into relation with initial parameters and give insights into their influence on the simulation.

Finally, in ongoing projects we dive into the recently emerging field of deep neural networks, where two directions are explored in more detail. Weather simulations are frequently run on terrain following grids but investigated at certain pressure levels requiring interpolation. In the first project, the ability of convolutional neural networks to interpolate between levels in physical scalar fields such as temperature are investigated. In the second, we explore how these networks can construct dense fields from sparse input, where some of the input is only visible because of the sparsity, which the network has to learn additionally. We applied this to sparse 3D point cloud renderings, where each projected point was rendered as a single pixel which could be used later for combining, correcting and completing meteorological measurements, or for rendering particle based simulations in an efficient way.

Zusammenfassung

Numerische Simulationen sind von Natur aus sensitiv gegenüber Eingabeparametern, welche meist nur geschätzt sind. Durch wiederholtes Simulieren mit verschiedenen Parametrisierungen wird ein Ensemble von Simulationen erstellt, um eine Repräsentation des Raums der möglichen Simulationsergebnisse zu erhalten. In den meisten Fällen ist es nicht praktikabel und zu zeitaufwändig, jede Simulation separat zu untersuchen. Daher wird für gewöhnlich nach Trends und Ähnlichkeiten im Ensemble gesucht, was allerdings zu neuen Unsicherheiten führt. Abhängigkeiten zwischen verschiedenen Variablen zu unterschiedlichen Zeitschritten können Hinweise liefern, welche Fehler sich über die Zeit ausbreiten und wo zusätzliche Messwerte die Simulation verbessern könnten. Um Unterschiede ganzer Simulationen zu quantifizieren, werden Metriken benötigt. Dadurch können Rückschlüsse auf den Einfluss einzelner Anfangsparameter auf die Simulation gezogen werden.

Mit Hilfe von Clustering können Ensemble Member gruppiert werden, was jedoch abhängig von den genauen Clustering Parametern unterschiedliche Ergebnisse liefert. Durch systematisches Ändern der Parameter wird ein Ensemble generiert, welches auf bestehende Sensitivitäten hinweist. Mehrere verzahnte Ansichten unterstützen den Nutzer bei der Auswahl eines geeigneten Clusterings und zeigen dessen Qualität und Eigenschaften. Übersichtsdiagramme zeigen die Variabilität in Clustergrößen, Clusterzugehörigkeit aller Datenpunkte und Unsicherheiten in Featurevisualisierungen wie beispielsweise Konturbändern. Durch Selektion eines bestimmten Clusters oder Clusterings werden alle Diagramme interaktiv aktualisiert. Genannte Methoden wurden von uns in einem Arbeitsablauf zusammengefasst und beispielhaft auf eine Vorhersage des Zyklons Karl im Jahr 2016 angewandt.

Korrelationen weisen auf Abhängigkeiten zwischen verschienden Regionen, Variablen oder Zeitschritten hin. Auftretende Korrelationsstrukturen sind dabei abhängig vom initialen Ort, zu dem korreliert wird. Zudem müssen diese über die Zeit einander zugeordnet werden. Durch Clustering von Gridpunkten werden statistisch kohärente Strukturen bestimmt, zu welchen anschließend andere Felder korreliert werden. Hinweise auf die Ähnlichkeit von Strukturen im Ensemble an einem Ort können durch Clustering hoch-korrelierter Regionen gefunden werden. Mit Hilfe von bi-direktionalem optischem Fluss werden Strukturen automatisch getrackt und in einem Split-Merge Diagramm zusammengefasst. Für einen selektierten Pfad mit ausreichend guter Trackingqualität wird ein Übersichtsplot erstellt, der die Struktur mit seiner zeitlichen Änderung in der Domäne zusammenfasst. Im oben genannten Anwedungsfall konnte eine Beziehung zwischen der Unsicher-

heit in der Position des Sturms Karl und einem nicht vorhergesagten Starkregenereignis in Norwegen einige Tage später festgestellt werden.

Um geeignete Parametrisierungen zu finden muss der Einfluss einzelner Parametern auf Simulationsergebnisse bekannt sein, wofür Simulationsobjekte als gesamtes verglichen werden müssen. Da sich diese räumlich ändern können, versagen ortsabhängige Maße, weshalb wir uns stattdessen auf Multiparameterverteilungen konzentrieren. Unterschiede deren Verteilungsfunktionen im Ensemble dienen zur Identifikation ähnlicher Strukturen in verschiedenen Membern. Relevante Strukturen werden durch Kombination mehrerer k-Means Clusterings auf t-SNE Projektionen identifiziert. Parallele Koordinaten dienen zur zeitlichen Darstellung und aufsummierte Differenzen der Verteilungsfunktionen liefern Rückschlüsse auf den Einfluss der Anfangsparameter.

In fortlaufenden Projekten beschäftigen wir uns mit Dateninterpolation und Datenvervollständigung mit Hilfe von neuronalen Netzen. Wettersimulationen werden oft auf orographiefolgenden Gittern berechnet, aber auf Leveln von konstantem Druck analysiert, wofür Werte interpoliert werden müssen. Anhand von Temperaturfeldern zeigen wir das Potenzial neuronaler Netze für Interpolation auf. In einem zweiten Projekt untersuchen wir, inwiefern neuronale Netze Löcher in Daten auffüllen können. Dazu korrigieren und vervollständigen wir in einem ersten Schritt Renderings von 3D Punktwolken. In einem nächsten Schritt könnten mit der selben Technik meteorologische Messdaten zusammengefügt und vervollständigt werden.

Acknowledgments

Before all others, I would like to thank Prof. Dr. Rüdiger Westermann for advising and supervising me during my PhD time. While always being available for his students, he suggested promising directions without restricting me in any way. Discussions were always honest and constructive. Moreover, he introduced me to the scientific community at conferences and to people from all over the world. His unsurpassable supervision, funny remarks and joyful chats with him made my PhD time just fly by.

Special thanks also to Dr. Marc Rautenhaus, who guided me throughout my PhD. He was always available and helpful, bridged the gap to meteorology and pointed me into interesting research directions. Also our camping trip around Grand Canyon was awesome!

Sincere thanks also to my other co-authors, PD Dr. Michael Riemer, Bianca Tost, Josef Stumpfegger, Patrick Härtl, Christian Reinbold and Marlene Baumgart, who always contributed timely and were not shied away by last minute requests.

Rahel Mengyu Chu was the best office mate one could wish for. She had the answer to all my neural networks related questions and discussions about other scientific problems were always fruitful. I will miss our conversations and all the fun time we had, including our plant surviving contest.

Susanne Weitz and Sebastian Wohner removed all organizational obstacles for me, starting from broken computers, over all kinds of forms and printing oversized posters. I thank Prof. Dr. Filip Sadlo, my second mentor, for lively discussions and advice, Prof. Dr. George C. Craig for his continuous support of visualization within W2W and Dr. Audine Laurian, Andrea Schäfer and Dr. Robert Redl for their organizational help, and all other members and PhD students of W2W.

My colleagues and friends Dr. Ismail Demir, Dr. Marie-Lena Eckert, Martin Ender, Dr. Florian Ferstl, Seyedbehdad Ghaffari, Kevin Höhlein, Philipp Holl, Dr. Mihaela Jarema, Dr. Mathias Kanzler, Dr. Johannes Kehrer, Michael Kern, Ludwig Leonard, Henrik Masbruch, Lukas Prantl, Dr. Yvonne Ruckstuhl, Moritz Schöpf, Prof. Dr. Nils Thürey, Junpeng Wang, Sebastian Weiß, Steffen Wiewel, Dequan Yang and You Xie and all the ones not mentioned were decisive in making my PhD time unforgettable.

My family and Heike Glasenhardt always provided me with everything I needed throughout my whole studies and gave me stability in stressful times. I am eternally grateful for that.

The research leading to these results has been done within the subproject B5 of the Transregional Collaborative Research Center SFB/TRR 165 Waves to Weather funded by the German Research Foundation (DFG).

Contents

1 Introduction

In numerical weather prediction a chaotic system is modeled and simulated. These simulations are commonly conducted in 3D over many time steps and multiple physical variables such as pressure, temperature and precipitation, leading to huge data sets. While the predictive power of such models has tremendously increased over the past decades, these models still can only approximate the real world. There are many processes involved, which are either not fully understood or are computationally too expensive to simulate properly, such as micro-physics in clouds. Systems for measuring physical quantities like temperature can only do so up to a certain precision and measurements can be missing due to occlusions or the lack of measuring stations. Discretization of continuous effects and limited simulation resolution additionally hinders more precise predictions. These uncertainties have to be estimated to better convey the expected quality of a forecast at certain locations.

By using modified initial conditions or assuming different parameterizations, the simulation can be re-run and alternative outcomes are generated, leading to an ensemble of forecasts, where each ensemble member predicts one possible future state. Even though the space of all possibilities conditioned on the knowns is under-sampled, the ensemble provides a better understanding of the possible deviations from a forecast run with the best initial conditions known.

Analyzing every ensemble member by its own is unfeasible; hence, methods to put them into relation have to be used. Starting by computing mean and standard deviations over the ensemble per grid location, regions of high uncertainty and major trends can be found. By clustering the ensemble members based on certain regions, similar forecasts can be detected and a more detailed analysis can be performed on few representative members. For instance, the European Centre for Medium-Range Weather Forecasts (ECMWF) operationally applies dimensionality reduction on a region defined as Europe and uses k-Means clustering to provide major trends and cluster representatives [FC11].

Depending on the weather situation, clustering can be sensitive to changing clustering parameters, such as the exact region used, the accuracy of compression after dimensionality reduction, or the

final number of clusters used. Visualizations using resulting variations in clusterings should reveal how much confidence can be put into single clusters and which members where clustered together consistently. Also, the fit and benefit of a chosen clustering on different regions has to be considered as well as uncertainties in feature visualizations introduced by different clustering results.

The ensemble contains different possible outcomes at all locations for all simulation times. Many quantities influence others at later times, e.g., wind transports moisture over time. Correlation analysis can hint on dependencies within the ensemble which can indicate dependencies in the real atmospheric dynamics. Ensemble values in one quantity at one location can be correlated to the values at a second location of another variable. Theoretically, this can be done for all locations making the computation unfeasible. Further, useful patterns still would have to be extracted from all these correlation values. In meteorology, ensemble sensitivity analysis (ESA) introduced by Ancell and Hakim [AH07] and Torn and Hakim [TH08] is used instead, correlating a region of interest to one other quantity at a time. ESA differs from Pearson's correlation coefficient by a scaling with the standard deviation to express correlations in terms of units instead of standard deviations. Now, instead of basing the correlation analysis on the values at a single grid location, regions are summarized by for example computing the mean or root-mean-square-error over the grid-points contained. This leads again to a vector with one value per ensemble member which can be correlated to all locations of the same or another scalar variable. Since the region of interest, for example a region of high precipitation, is at a different location in every ensemble member, this one region cannot be perfect for all members at once; hence, methods for detecting statistically coherent regions are needed which further indicate, where the analysis is valid. In a next step, this correlation computation can be done for multiple time steps and areas of high correlation can be tracked over time possibly leading to the origin of uncertainty for the region of interest. This process has to be repeated for different initial regions, so an automatic tracking is desirable and the whole time evolution has to be summarized for a first and fast overview. Further, it is worth knowing whether found correlation regions represent coherent phenomena in the ensemble.

When relating differences in simulations even further to the initial parameters, different methods are needed to measure similarities of whole simulations and time steps. By separately simulating single structures such as a single growing cloud, the exact location and size of the object can change calling for location invariant methods. Feature extraction and comparison only works if relevant features are known. Investigating the distributions of simulated parameter values of simulation elements instead can serve as an alternative. Either all simulation elements are compared at once or substructures are searched and compared to those in other time steps or members for a finer comparison. Suitable measures have to be applied for this comparison and a matching between the structures has to be performed. Finally, correlations to initial parameters can be computed to reveal the effects of parameter perturbations on the simulation.

Interpolation is constantly used when regridding, slicing through volumes or estimating intermediate time steps. Hence, an accurate interpolation method is needed to achieve good results. The problem of interpolating intermediate frames has been discussed extensively in the context of video frame interpolation. There, the best results are achieved using deep neural networks, which either predict the intermediate frame directly, or are based on optical-flow-based warping of the enclosing frames. We investigate how these methods perform when interpolating vertical levels in numerical weather forecast data. As scientific data, such as temperature fields, has less pronounced edges than cars or persons in videos, and further, the value range and distribution change permanently, it is worth investigating to what extend the same approaches work there.

During data assimilation, occlusion effects lead to missing values, which have to be filled. When combining measurement data from different sources, inconsistencies in the data occur. In image processing, the first task is referred to as inpainting, where missing pieces are reconstructed using the available context data. Convolutional neural network based inpainting approaches offer a more realistic gap filling than classical approaches by taking pre-trained information from other images into account as well. Enough realistic training data is important to achieve plausible results, which is why we explore a slightly different but closely related task first. We render a sparse version of a 3D point cloud as pixel splats and let the neural network fill in the holes and remove points from occluded objects which only became visible due to the sparsity of the data. The used point clouds contain scans from historical buildings where each point is attributed with color and the normal of the surface. This data is potentially better posed than measurements of physical quantities and therefore is a good starting point to assess how neural networks could contribute there. Both neural network projects are at the time of writing unpublished and ongoing work with promising first results.

1.1 Contribution

In the context of this thesis contributions for different problems have been made. Clustering is often used without a proper analysis of its uncertainties. In the context of a workflow, we present means to compare different clustering results in terms of cluster size, membership and gain from clustering. Correlation analysis in ensemble forecast analysis is improved by assisting the user in initial region selection giving interactive feedback on resulting correlations, automated tracking of correlation structures and overview plots to navigate and summarize correlation structures. Based on clustering and cumulative distribution functions, we relate whole simulations to their initial parameters and provide a method to identify similar structures over different time steps and ensemble members. In unpublished work, we show how neural networks can be used to improve interpolation between different layers in data. Further, we show their ability to repair and complete erroneous renderings of 3D point clouds. Proposed methods are tailored to work well for numerical weather

forecast data, however, most of them are not limited to it. More specifically, structured by publication, the following contributions were made:

- We present a workflow to assess the confidence which can be put into a clustering. Once an ensemble of possible clusterings is generated, a cluster-centric robustness display (CRD) conveys information about cluster compositions, changes in cluster amount and sizes as well as to which clusters changing members belonged in other clusterings. A member-centric robustness display (MRD) encodes in one glyph per member the fractions of how often that member belonged to each cluster. More detailed information is shown in pop-up matrix glyphs and similarity queries help finding members with similar member-change characteristics. The variance reduction achieved by a selected clustering at each grid point is computed and variations in geographical plots are indicated using texture stippling. A novel contour plot developed by Tost and Rautenhaus is presented which can be attributed with this stippling as well. An interactive interface enables the synchronization and coupling of all plots, where single clusterings, clusters, members and manually chosen groups can be selected and all views are updated accordingly. In a following case study, the tool is used to find a suitable clustering for a data set forecasting tropical cyclone Karl in 2016, also indicating regions which could not be clustered well by the chosen algorithm. (see [Kum+18])
  As it turned out, the combination of k-Means clustering and MRD with its functionality can be used to investigate neighborhood stability in ensembles of 2D point clouds in general. (see [RKW19], not part of this thesis)
- Common practice in ESA is improved by providing a statistically meaningful initial region to a selected seed-point, to which a second quantity is then correlated. This way, mainly grid points leading to consistent results are used and shown, and others potentially worsening the analysis are excluded. Using an automated optical-flow-based forward-backward tracking, correlation structures are traced over time and summarized in one plot. A split-merge diagram is used to control filtering of tracked paths by their tracking confidence and to select single tracked structures. Single correlation structures are clustered into coherent spatial regions to indicate the extend with similar grid points in the ensemble. In a case study, a link between a heavy precipitation event in Norway to the uncertainties in the moisture distribution around tropical cyclone Karl was found. (see [Kum+19])
- We present a distribution based similarity metric to compare sets of simulation elements. By combining multiple k-Means clusterings on t-SNE projections of a set of simulation elements, a stable clustering for arbitrarily shaped clusters is achieved. Based on this, relations between perturbations to initial parameters and differences to a chosen reference data set can be found and investigated. (see [KSW19])

- In unpublished work, deep convolutional neural networks such as the U-Net or EnhanceNet are used to interpolate intermediate levels with higher accuracy than linear interpolation. The same architectures are able to complete and correct pixel-splat renderings of sparse 3D point clouds.

1.2 List of Publications

A. Kumpf, B. Tost, M. Baumgart, M. Riemer, R. Westermann, and M. Rautenhaus. "Visualizing Confidence in Cluster-based Ensemble Weather Forecast Analyses". In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 109–119

A. Kumpf, M. Rautenhaus, M. Riemer, and R. Westermann. "Visual Analysis of the Temporal Evolution of Ensemble Forecast Sensitivities". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 98–108

A. Kumpf, J. Stumpfegger, and R. Westermann. "Cluster-based Analysis of Multi-Parameter Distributions in Cloud Simulation Ensembles". In: *Vision, Modeling and Visualization*. The Eurographics Association, 2019

Further publication not part of this thesis

C. Reinbold, A. Kumpf, and R. Westermann. "Visualizing the Stability of 2D Point Sets from Dimensionality Reduction Techniques". In: *Computer Graphics Forum*. Wiley Online Library. 2019

2

Related work

This section is designed to provide the reader with literature relevant for understanding the context and methods presented in our papers. While referenced work overlaps with the one presented in our papers, more detailed descriptions of the contents are given here in order to help the reader determine which articles to read. For a more exhaustive discussion of literature available in each topic, we refer to the state-of-the-art reports mentioned.

Uncertainty visualization Uncertainty visualization is used to adequately communicate possible deviations of data. As this thesis focuses on ensemble data, these methods are constantly used in our work.

In 2003, Johnson and Allen [JS03] raised awareness for the necessity of visualizing uncertainty by presenting an overview of uncertainty visualization methods which lead to a better understanding of the data. Discussed sources for uncertainty are measurement errors, compression for storing, running models which only imitate the real world, transformations and interpolation, and lastly also the visualizations themselves.

Not much later, Thomson et al. [Tho+05] gave a thorough updated overview of different sources and types of uncertainty and present a typology to better describe its characteristics. Griethe and Schumann [G+06] provide an overview of uncertainty visualization in general together with some early methods which can be used. Uncertainty can be mapped to unused graphical channels like color, size, position or transparency. Alternatively, small graphical objects such as glyphs, labels or error bars can be added to existing visualizations, which is still common practice today. Interactivity can help the user to browse through the data, while animations show changes in the data with the disadvantage that the user has to remember previous configurations. Further, the order in which they are shown changes the perception of the data.

More recently, Potter et al. [PRJ11] summarized works with the focus on scalar, vector and tensor data uncertainty, while Bonneau et al. [Bon+14] focus more on the types of uncertainty and offer guidelines and small examples. Li et al. [Li+07] point out that the visualization of uncertainty should not occlude and distract from the actual data. They use error bars and ellipsoids to convey uncertainty present.

Among the most used techniques are box plots presented by Tukey [Tuk77], which are still used today because of their simplicity. A line in a box marks the median, whereas the upper and lower boundaries mark the first quartile of the data. The third quartile is included using whiskers and outliers further away can be included as small dots. Other established methods used for clustered climate data are discussed by Nocke et al. [NSB04], including ThemeRivers [HHN00], scatter plot matrices [TT88] and parallel coordinates [Ins85].

Glyphs, small geometric objects which can encode multiple values, are also suited to encode uncertainty information. Due to their small size, they can be used in a visualization at multiple places at once encoding relevant information there. Ropinski et al. [ROP11] summarize existing glyph designs used in medical visualization and give usage guidelines based on their literature review. The right placement of glyphs is crucial to convey information appropriately. Ward [War02] provides an overview over existing placement strategies while considering additional constraints such as allowed overlap, screen space utilization or the option for manual placement adjustment. More recently, Borgo et al. [Bor+13] summarized many works, with an additional focus on design criteria for glyphs.

Uncertainty of vectors such as gradients can be modeled using mixed Gaussian distributions as done by Pfaffelmoser et al. [PMW13]. They encode local gradient strength and variability with a diffusion texture and a multi-bin color scheme. Uncertainties of gradient directions are shown using circular glyphs.

Ensemble visualization Closely related to uncertainty visualization is ensemble visualization. Ensembles contain multiple version of the same object, such as the values at a grid location, a feature or even the whole simulation. On the one hand, these can be used to compute an estimate for the uncertainty of a simulation element, but can also be interpreted as multiple possible realizations on the other hand. Different types of objects require different visualization methods. Obermaier and Joy [OJ14] divided the field of ensemble visualization into the two groups of feature-based and location-based methods, where methods of both groups have been used in the work leading to this thesis. One example for feature-based ensemble visualization is given by Demir et al. [DKW16], where they generate screen space silhouettes to visualize an ensemble of iso-surfaces.

Love et al. [LPK05] recognized the importance of such visualizations and propose techniques to visualize multi-value data sets, including ensemble data. For data on a 2D domain, basic approaches as mapping mean values to color, encoding the standard deviation into saturation or a height map, or

using spaghetti plots to show the range of single iso-contours in an ensemble are investigated. Also other surveys summarize relevant work in the field. While Kehrer and Hauser [KH13] summarize works depending on their data type, i.e., multi-modal, multi-run and multi-model data, Wang et al. [Wan+18] focus on recently proposed ensemble visualization techniques and categorize them according to the analytical task they can be used for.

Rautenhaus et al. [Rau+17] summarized visualization methods used in meteorology, including visualization tools, uncertainty, ensemble and feature visualization methods. The ensemble tool Met.3D developed by Rautenhaus et al. [Rau+15a; Rau+15b] is tailored for visualizing ensemble data and contains many basic visualizations, such as mean and variance computations, feature extraction as well as cutting edge methods for research analysis such as the detection of weather fronts [Ker+18a] and jet cores [Ker+18b]. Met.3D provided a suitable infrastructure for comparing clusters [Kum+18] and investigating correlations [Kum+19], as its pipeline data-management handles caching and data retrieval to enable interactive exploration.

Many ensemble data sets also contain a time component, which has to be taken into account. Aigner et al. [Aig+07a; Aig+07b] summarize works focusing on visualizing, summarizing, abstracting and interacting with time-dependent data. Ferstl et al. [Fer+17] investigate the divergence of clusters in an ensemble over time, and Leistikow et al. [Lei+19] use function plots to encode the evolution of members over time. Jarema et al. [JKW16] use multiple linked views to visualize the flow transport variability in ensembles. By fitting Gaussian mixture models (GMMs) to particle positions seeded at one location and time in the ensemble and evaluating their similarity using the Mahalanobis distance, spaghetti plots are color-coded. Small multiples provide an overview for the whole domain and a cluster split-merge graph depicts the variability of cluster sizes over time.

Coordinated views Especially ensemble data contains lots of information which oftentimes cannot be displayed in a single visualization. A now common approach is to use multiple visualizations simultaneously instead, and interconnecting them to visualize different aspects of the same domain or show context information.

Fofonov et al. [FML16] compare iso-contour similarity by checking for random locations whether values are within or outside the contour. The resulting binary vector can be compared with others indicating similarities in location and shape, and further, dimensionality reduction is applied to visualize the spatio-temporal evolution of multiple iso-contours simultaneously. The views are linked to be able to relate the visualizations with each other. Fonfonov and Linsen [FL18] use aggregated plots and coordinated views to investigate an ensemble data set. Histograms are used to show value distributions and data ranges where color encodes how often the ranges occur in other runs. A function plot shows variation of members over time and and a similarity plot encodes where members diverge in time. Leistikow et al. [Lei+19] use multiple coordinated views to investigate a time-dependent

ensemble data set by combining function plots to represent each member as a time-series graph, multi-run similarity plots to encode iso-surface similarity over time and volume visualizations.

Contour plots Iso-contours are especially important features in meteorology. For example, the iso-contour in the potential vorticity unit field – a derived variable representing the dynamics in the atmosphere– marking a level of 2 can be used as a representation of the beginning of the tropopause. Errors in the weather forecast in the exact shape of this iso-contour are often correlated with the overall quality of a forecast, which makes it especially useful to investigate [Bau+18]. In the case of ensembles of 2D scalar fields, iso-contours can be extracted for every ensemble member, leading to so called spaghetti plots. A different color can be used for every ensemble member, but eventually the plot becomes confusing and major trends are not visible anymore. Multiple methods have been proposed to tackle this problem. There is a distinction between parametric and non-parametric methods, where the non-parametric ones do not assume the lines to underlie a specific distribution. For example, Sanyal et al. [San+10] proposed the use of confidence ribbons. Regularly spaced circular uncertainty glyphs with uncertainty mapped to radius are generated along a contour line. By connecting tangents between adjacent circles, confidence bands are generated. Pfaffelmoser and Westermann [PW13] visualized the distribution of iso-contours without assuming them to be normally distributed by computing a spatial cumulative distribution function field over the ensemble. Contour boxplots introduced by Whitaker et al. [WMK13] are based on band depth measuring the percentage of contours contained within. Shown confidence regions then use existing contours as borders. Further, outliers are marked in red to display the maximal deviation present as well. As an extension, Mizargar et al. [MWK14] proposed curve boxplots which can handle arbitrary curves instead of lines which are either closed or traverse the whole domain such as iso-contours. Parametric models on the other hand assume certain distributions. Ferstl et al. [FBW16] proposed variability plots, which transform lines of equal length into lower-dimensional Euclidean space using principal component analysis (PCA). The lines are then optionally clustered and multivariate normal distributions are fitted onto them. The median and confidence ellipses at a certain standard deviation level are then transformed back representing the center and outer contours of the confidence band. In a subsequent work [Fer+16; Fer16], the method was improved using signed distance fields (SDF) on the initial iso-contours before the PCA eliminating the restriction to lines having the same length. Probabilities of lines passing through certain locations conditioned on the passing through a defined circle is achieved by intersecting slaps in the signed distance space.

In our work [Kum+18], Bianca Tost and Marc Rautenhaus developed contour probability plots, which are based on a field showing the grid-point wise probability of exceeding an iso-value in the ensemble. Iso-contours at different probability levels then represent the confidence regions' borders. Uncertainty in the position of the confidence bands, when an ensemble of spaghetti plots is available,

can be visualized by encoding the variance in probabilities using texture stippling.

Clustering and cluster analysis in meteorology Clustering can be used to group objects together according to a similarity criteria. It has been applied to numerous disciplines, including meteorology, where a general overview of statistical methods and clustering algorithms used can be found in the book of Wilks [Wil05]. Hereinafter, the most relevant techniques to our work are summarized.

Agglomerative hierarchical clustering [War63] builds up a clustering hierarchy by successively merging close objects into the same group until only one cluster remains. When stopping after a certain similarity value, multiple clusters with a minimal distance greater than the value are obtained. Ferstl et al. [FBW16] transform streamlines of equal length, or streamlines represented as signed distance fields [Fer+16], with principal component analysis first, and then cluster in the transformed space with hierarchical clustering. In subsequent work [Fer+17] they generate contour plot visualizations of clusters found in the last time step, stack them vertically for different time steps and display cluster merge events for earlier time steps–identified by cluster distances falling below a threshold– using a split-merge diagram. Recently, Kern et al. [KW19] compare different clustering approaches for line clustering and evaluate which method leads to a good representation of line features contained in a cluster.

While hierarchical clustering algorithms are especially useful for finding elongated structures, the results can be very sensitive to changing data. Additionally, convex cluster structures are better handled by k-Means. Lloyd's k-Means algorithm [Llo82], which aims at placing a predefined number of cluster centers in a way that the sum of squared distances of all objects to their center is minimized, is one of the most commonly used. Numerous improvements of it have been proposed, such as k-Means++ [AV07] using an improved seeding strategy by better distributing seed cluster centers, or k-Medians, were the cluster center is the median of the points contained, which is more robust to outliers.

Frame et al. [Fra+11] for example use k-Means to cluster jet wind profiles which are then compared to 23 years extended winter climatology. Most relevant to the clustering part in our first paper [Kum+18] is the work by Ferranti and Corti [FC11]. They use k-Means clustering to group ensemble members into major trends based on a region defined as Europe. More details are given in Section 3.1.5.

All clustering algorithms have in common that their results depend on the input parameters and data, and already small perturbations thereof can lead to significantly different results. There are two approaches to deal with this. A clustering algorithm should always find a result close to its optimal clustering criteria. Hence, a clustering can be both optimal but sensitive to the chosen parameters. Our first work aims at visualizing these sensitivities to make the user aware of possibly different outcomes and leave it to the user to decide whether and which clustering to use [Kum+18].

When it is clear that a single clustering cannot capture all clusters correctly, as it is the case of clustering in t-distributed stochastic neighbour embeddings (t-SNE) [KSW19], clustering multiple times instead and combining these can lead to the desired final clustering. Strehl and Ghosh [SG02] discuss how to combine the results of different clustering algorithms into a final one. More recent techniques are discussed by Vega-Pons and Ruiz-Shulcloper [VR11]. The technique we use is close to the work of Fred and Jain [FJ05], who use evidence accumulation by applying single-link or average-link hierarchical clustering on the co-association matrix. In contrast, we apply greedy region growing on cliques of similar data-points concerning their cluster memberships.

We chose the indirect clustering step using t-SNE projections [MH08] beforehand to make use of its objective to preserve local neighborhoods of points. However, also other algorithms such as density based clustering, e.g., DBSCAN by Ester et al. [Est+96] could be used but did not lead to the same quality of results. Another possibility is offered by Molchanov and Linsen [ML18a; ML18b], who cluster arbitrarily shaped clusters by subdividing the multi-dimensional parameter space successively into bins. Bins with density below a predefined threshold are ignored to remove noise and speed up the algorithm. All neighboring bins without a gap are considered as a cluster. By successively increasing the noise threshold, connections break as bins are removed and a cluster hierarchy is generated. To lighten the impact of chosen bin sizes, interpolation is used to increase the number of high-dimensional sample points.

Dimensionality reduction High-dimensional data comes with several disadvantages. With every additional dimension, distances become harder to distinguish, memory requirements increase and visualization becomes more challenging. Several algorithms exist for mapping data into a lower-dimensional space, each having different benefits. Principal component analysis, explained in detail in the book by Jolliffe [Jol10], applies an orthogonal linear transformation to the original data to maximize the data's variance in one dimension after the other. It can be implemented efficiently using singular value decomposition. By neglecting all dimensions but the first k in the transformed space, the data is projected by preserving the maximal amount of variance possible.

T-distributed stochastic neighbour embedding (t-SNE) presented by Maaten and Hinton [MH08] aims at preserving local neighborhood information by computing neighborhood probabilities in the high-dimensional space based on normal distributions. Measuring the similarity in low-dimensional space using a Student t-distribution and minimizing the Kullback-Leibler divergence with gradient decent leads to the final projection. A random initial configuration is used on which this method depends, which is why for different random fields different projections are achieved, in contrast to the fully deterministic PCA. Other projection algorithms such as Multi Dimensional Scaling [KW78] are summarized in a survey by Sorzano et al. [SVM14]. To improve the quality of a projection, Molchanov and Linsen [ML14] let the user steer the projection by moving control points and optimizing iteratively

an overdetermined least squares problem. The user can slide through the optimization steps and stop at any point, while star coordinates are shown to help understand the changes made.

In our work, we apply PCA [Kum+18] and use the variation in t-SNE to extract arbitrarily shaped clusters [KSW19] in high-dimensional data. Together with Reinbold et al. [RKW19], we investigated the stability of 2D point sets, which can also originate from dimensionality reduction.

Distribution analysis Distributions can be used to model uncertainty or compare different sets. Numerous works fit distributions per grid-location to model variation in ensemble data. The mean and standard deviation [LPK05] are easy and fast to compute and completely describe a normal distribution, however, they model the data only roughly in most cases.

A common enhancement is modeling distributions using GMMs. Liu et al. [Liu+12] fit GMMs to compress the ensemble vector at every grid location, reducing the values to store to the GMM parameters. Wang et al. [Wan+17] compute histograms per block and model the grid-points contained in each bin using GMMs. Using Bayes' rule, values at locations can be reconstructed.

Jarema et al. [Jar+15] model variations in vector field ensembles by mixtures of probability density functions and map them onto vector glyphs. Dutta and Shen [DS15] identify structures by fitting GMMs to blocks of the data using expectation maximization. Under the assumption that distributions in the same region only change slightly over time, only parameters of some Gaussians are replaced at successive time-steps instead of re-estimating the whole GMM. Assuming the background distribution to stay similar, changing parameters indicate features passing through the blocks and are then used for the tracking of such.

When no assumption about values following a certain distribution should be made, non-parametric models are used. To represent these, Chambers et al. [Cha+83] visualize the distribution of values as a continuous line and Hintze and Nelson [HN98] combine box plots and trace plots to violin diagrams. Hazarika et al. [HBS17] use copulas to model distributions and dependencies of the distributions of nearby grid points, which are common in scientific data. In following work [Haz+19], they use copula functions to efficiently compress data and demonstrate its use in an in situ environment.

For more basic representations such as empirical distributions, histograms or kernel density estimates, Pöthkow and Hege [PH13] extend existing feature extraction techniques, and show their benefit on level crossing probabilities amongst others. Athawale et al. [ASE15] explore the extraction of iso-surfaces on fields with non-parametric distributions. Demir et al. [DDW14] create many side-by-side color-coded bar-histograms to encode value distributions of different spatial areas in ensemble data. The order of the histograms is determined by a space filling curve in the data set and 3D context is retained by linking brushed histograms with highlights in the spatial view. The spatial bin size for the histograms can be adjusted to get a more detailed analysis of the value distributions.

Correlation analysis and correlation clustering Analyzing and visualizing correlations in 3D scalar field ensembles is challenging due to the huge amount of correlation values which can be computed, using for example Pearson's correlation coefficient. Visualizing them in correlation matrices only works for small data sets. When it is still feasible to compute them, clustering on the correlation matrix is used to reduce the data complexity, for which various algorithms exist. Bansal et al. [BBC04] cluster graphs maximizing pair-wise similarities– which can be positive correlations– while minimizing dissimilarities within clusters. While their approach is restricted to the values $\{+1, -1\}$, Demaine and Immorlica [DI03] extend the idea to arbitrary weights and propose an approximation algorithm using region growing. Recently, Liebmann et al. [LWS18] proposed to use hierarchical clustering on distances of correlation data which was transformed onto hyperspheres.

While these approaches work for arbitrary correlation relations, methods operating on grid data are more relevant for our work in [Kum+19]. In early work, Jen et al. [Jen+04] explore correlations between two 3D scalar fields at the same location by using 2D slices and mapping one variable to height and the other to color. Additionally, values along trajectories are displayed in a line chart.

Oftentimes, values in a certain region are interpreted as a sample of a random variable, or in case of ensembles, the ensemble values at a single grid location. Sauber et al. [STS06] focus on the visualization and navigation through correlation fields, where correlations at the same location but different subsets of the ensemble are considered. Correlations between locations, different variables and time steps are investigated by Sukharev et al. [Suk+09]. They further cluster time-activity curves [Fan+07] using k-Means to reveal patterns in the data.

To reduce the number of correlation calculations, Chen et al. [Che+11] present a sampling based approach operating on importance maps provided by domain scientists. For every sample, a correlation volume is computed. The distance between those volumes is then estimated by comparing histograms of correlation values contained using the Jensen-Shannon divergence measure [Lin91].

Pfaffelmoser and Westermann [PW12] cluster regions of high correlation without requiring domain knowledge. Grid points are sorted according to the size of the connected region around them with positive correlation values above a threshold. The highest ranked point defines the largest region which is the first cluster. Iteratively, successive largest regions define further clusters with the additional constraint of disjointness of points contained, clustering the 2D domain into non-overlapping clusters. Clusters are further subdivided using the same algorithm. The effectiveness of the method is demonstrated on an 2D scalar field ensemble.

Recently, Antonov et al. [Ant+19] presented a tool to interactively investigate teleconnections of correlations in climate simulations. By iteratively searching for strongest anti-correlations between a starting grid point and all others, a line-strip is generated to show correlation teleconnections in the domain. At each point of the line-strip, region growing is used to partition the domain into correlation areas. Coordinating this view with a projected view of the line strip, an interactive tool is presented.

In our approach [Kum+19], cliques of points with high pair-wise auto-correlation in the ensemble are determined for a user-selected location. The values of these points are then summarized, e.g., by taking the mean or mean-squared-error from a given measurement, resulting in one scalar value per ensemble member. This vector is then correlated to all locations of a selected second variable at a specific time, and highly correlated regions are again clustered into disjoint cliques as described before by starting with the location showing highest correlation. The initial clique provides a region of points leading to similar correlation patterns and guides the user by showing for which region the analysis is valid. The clustering of correlation structures in the second variable gives hints about the regional consistency of patterns in the ensemble.

Tracking of structures and optical flow When the same structures appear over multiple time steps, they can be matched using tracking algorithms. While tracking in computer vision is often based on symmetries and edges, tracking in visualization focuses more on the regions to track as a whole. Only works of the latter are discussed in the following.

Samtaney et al. [Sam+94] track features extracted from 3D volumetric data sets based on similar mass, distance of center, or spatial overlap— also used by Silver and Wang [SW97], or Sohn and Bajaj [SB06]—, which requires a high temporal resolution of the data to work properly. As an abstract representation for tracked structures, Bremer et al. [Bre+11] construct a tracking graph similar to ours, mapping attributes to nodes and offer the selection of single nodes. Sliders are used to filter structures to be shown based on, e.g., their minimal volume.

A well-arranged tracking graph minimizing crossings between lines can help to better understand the connections between structures. Even though it was not necessary in our case, let us refer to Widanagamaachchi et al. [Wid+12] for an efficient method to restructure a tracking graph.

A more elaborate tracking strategy is presented by Saikia et al. [SSW15; SW17], who track regions defined by merge trees. In the early work, they establish correspondence based on the L_2-norm on histograms of values contained in a feature. As an extension, all sub-tree pairs of two successive time steps are compared by weighting their overlap and value similarity computed by the Chi-Squared histogram distance [PW10]. They also experimented with the Earth Mover's Distance [PW09] instead, as Ji and Shen [JS06] do, however, this proved to have less discriminative power in their case. Finally, tracking is performed using Dijkstra's shortest path algorithm over all time steps to optimize the tracked paths globally.

Optical flow based methods additionally take into account the general directions in which structures are moving. First, the flow between successive fields is estimated, then the whole field or features are advected and structures are then finally matched according to some similarity criterion. An easy way to approximate the flow is by extrapolating existing tracks of already matched structures linearly or quadratically. Once tracks between two initial time steps are established, Muelder and

Ma [MM09] first translate features with their extrapolated track between feature centers and then apply the matching.

Using a dense pixel-wise estimation of the optical flow [Far03], Doraiswamy et al. [DNN13] and Valsangkar et al. [Ani+18] track cloud and cyclone structures performing an overlap test of advected regions to establish correspondence. Tracked regions are then blended onto each other with black outline contours and arrows between tracked cloud system centers.

As optical flow based tracking relies substantially on an accurate flow estimation, lots of research has been conducted to improve its estimation. Vanishing, changing and appearing structures especially complicate the task. Early methods were presented by Horn and Schunk [HS81] and Lucas and Kanade [L+81], whereas in [Kum+19], we use the more recent method of Farnebäck [Far03] implemented in openCV [BK00]. Optical flow methods are mostly designed to track structures which do not change too much over time in contrast to scientific data, where value ranges and distributions of the same structure can change and commonly no sharp borders are present. Nevertheless, the chosen algorithm performed well on our correlation structures.

Recently, deep neural network based flow estimation approaches have gained importance. FlowNet by Dosovitskiy et al. [Dos+15] uses an encoder-decoder U-Net [RFB15] architecture and FlowNet 2.0 by Ilg et al. [Ilg+17] improves it by applying multiple networks successively and optimizing the data scheduling for training. PWC-Net by Sun et al. [Sun+18] is smaller in size and outperforms FlowNet2.0 according to the authors. This is achieved by computing partial cost volumes, embedding them as layers into the network and refining the final flow by exploiting contextual information.

The optical flow can not only be used to track features but also to interpolate between frames in videos, or between slices in scalar fields (see Section 3.6). Many frame interpolation frameworks rely on a good optical flow map and then warp one or both images into the interpolated one.

Frame interpolation The generation of an intermediate frame using enclosing frames is an active field of study. It is used to increase the frame-rates in videos or computer games. In our project, we investigate how neural network based approaches known to perform well on videos work on scientific scalar fields, such as temperature or pressure in numerical weather forecasts. The abilities of neural networks on scientific data are already demonstrated by Weiss et al. [Wei+19], who present a super-resolution approach using a residual network architecture (see [He+16]) to improve isosurface renderings. Rasp and Lerch [RL18] showed that post-processing with neural networks can improve 2-m temperature forecasts.

There are three main approaches to interpolate intermediate layers. One can either interpolate between two enclosing frames, e.g., using linear interpolation, and let the network improve the interpolation based on the enclosing frames. The second approach aims at predicting the intermediate

frame directly without prior interpolation, while the third estimates the optical flow between the existing frames and warps each pixel to the intermediate time-step. Without going into much detail, the most relevant approaches inspiring us are briefly mentioned here. More comprehensive literature reviews to frame interpolation can be found in the respective papers. The use of the correct loss functions is crucial for obtaining good results; however, this is out of the scope of this thesis. For details, we again refer to the descriptions in the papers mentioned below.

Niklaus and Liu [NML17] use a U-Net architecture [RFB15] followed by four small convolutional sub-networks to predict the intermediate frame directly. In another approach [NL18], they estimate the optical flow between frames with the pre-trained PWC-Net [Sun+18], warp both enclosing frames and their context maps to an intermediate frame and use GridNet [Fou+17] to predict the final frame. Unfortunately, this network architecture proved to be difficult to train in our case.

Super SloMo by Jiang et al. [Jia+18] applies two U-Net networks successively, where the first is used to generate a rough estimate of the optical flow between the two enclosing frames, while the second U-Net refines it based on additionally added bi-linearly interpolated flows as well as forward and backward warped images to the intermediate time point. The outputs of the whole network are then refined flows to the intermediate frame as well as visibility maps correcting for occluded pixels. The advantage of this approach over others is that multiple intermediate frames can be generated with the flow afterwards.

A task related to interpolation is inpainting, where missing parts of an image have to be filled conditioned on the remaining context. Encoder-decoder networks are suited for this task, as they can incorporate context from a wide range of the image. Nazeri et al. [Naz+19] address the issue of over-smooth results in encoder-decoder networks by first completing and predicting edges in the image, before the actual colors are filled in. Liu et al. [Liu+18] use a U-Net architecture with partial convolutions instead. Partial convolutions deactivate neurons operating on missing values and therefore prevent them from spilling into the image.

Our task of improving sparsely rendered 3D point clouds is closely related to inpainting, as holes have to be filled based on surrounding pixels. Classical methods to visualize point clouds without constructing triangular meshes are discussed by Linsen [Lin01]. Due to the sparseness of point clouds, also structures behind objects can be visible in such renderings. The network has to detect and discard these. We achieved promising results by combining the ideas of Super SlowMo of two successive U-Nets. The first U-Net is used to predict which pixels of the sparse rendering are indeed visible while the second functions as an inpainting network.

3 Fundamentals and methods

The papers leading to this thesis contain contributions to different topics. This section is designed to provide some background information on the most important techniques used and additional details about our realizations are given.

3.1 Clustering analysis

The task of grouping objects based on a similarity criterion is called clustering. Representations of objects include geometric features, relationships, or coordinates in $\mathbb{R}^n$. A metric quantizes their similarity and rules for grouping objects lead to the final result. Numerous clustering algorithms exist, all with strengths and weaknesses of their own. An introduction to the topic is given in the book of Everitt et al. [Eve+11], together with the most commonly used techniques. In this chapter, the ones most relevant for our work will be discussed briefly.

3.1.1 k-Means clustering

One of the most used clustering algorithms for points in $\mathbb{R}^n$ is k-Means. It aims at placing a predefined number of cluster centers in the domain minimizing the sum of squared distances of all points to their closest cluster center, i.e., minimizing

$$\arg\min_{Cl} \sum_{j=1}^{k} \sum_{x \in Cl_j} ||x - \bar{\mu}_j||^2,$$

where k is the predefined number of clusters and $\bar{\mu}_j$ the mean of all points in cluster Cl_j, $Cl = \{Cl_1, \ldots, Cl_k\}$. Obtaining the minimum of the objective function of k-Means is known to be NP hard [MNV09]. One of the most used algorithms was introduced by Lloyd [Llo82], starting with

random cluster centers and then in each iteration, assigning every point to its closest center and re-computing the center as the mean of all points assigned. Since this algorithm terminates in local minima, it is re-run multiple times and the best result is taken.

Variants of this algorithm have been developed to speed up convergence. For instance, k-Means++ introduced by Arthur and Vassilvitskii [AV07] improves quality and convergence speed by a special placement of initial seed cluster centers. After randomly choosing the first center, successive centers are chosen with a probability proportional to the squared distances to the closest cluster center. A more even distribution of cluster centers is obtained which potentially improves convergence. Nevertheless, we used the Lloyd's algorithm in our paper [Kum+18]. The computational overhead for choosing the initial cluster centers exceeded its benefits since for this data only few iterations were needed for k-Means to converge to a local optimum. For accurate results, we re-ran k-Means multiple million times, especially for $k > 6$.

If cluster representatives are needed, either the point closest to its cluster center can be taken or the algorithm k-Medians can be used instead, where cluster centers have to resemble points of the data set. However, finding the median in a multi-dimensional data set is again a hard problem, leading to additional difficulties.

3.1.2 Hierarchical clustering

Hierarchical clustering algorithms create a hierarchy in which data points are combined to groups. For agglomerative hierarchical clustering, initially, all data points are assumed to lie in different clusters and the most similar clusters are merged successively until only one cluster remains. Contrary, divisive clustering starts using one cluster and divides it successively. Stopping this process at a certain similarity threshold or a predefined number of clusters leads to the final clustering result. Different metrics such as Euclidean distance or the Manhattan distance can be used to determine the similarity between points, while the linkage criterion decides on the actual clusters being merged. Common linkage criteria are complete or single linkage. This criteria defines which points of two clusters define their distance, i.e., the most distant or the closest points based on the chosen metric. The most similar clusters are then merged.

One advantage of hierarchical clustering is that it is deterministic. Outliers fall into their own cluster but few ill-placed points can lead to merges of clusters, which is why single-linkage often results in chainlike clusters, while complete linkage tends to break big clusters apart [Oel+14].

3.1.3 Density based clustering

A third group of clustering algorithms create clusters based on the density of points in the domain. The standard algorithm for this is Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

developed by Ester et al. [Est+96]. It is based on two parameters, namely the size ϵ of a neighborhood region and the minimal number of points *MinPts*, which have to be contained within the neighborhood to consider it a dense region.

The algorithm starts by selecting a random point and checking the number of neighbors in its ϵ-neighborhood. If it is below *MinPts*, the point is considered to be noise, otherwise all points contained are added to a list. For each point in the list, the same test is applied and new neighbors are added to it. Points on which the neighborhood query has been performed are removed from the list. Once the list is empty, the procedure is repeated with the next unprocessed point. The lists are numbered, and each point ever contained in the list is assigned its number as cluster ID. Further, every point is only processed once, but still counted in neighborhood searches.

The algorithm is almost deterministic. Points on the border of clusters can change cluster membership or even be considered noise when processed in a different order, but the overall clusters stay the same. By requiring only one pass through the data with a neighborhood search for a limited area around each point, it scales well for large data sets as long as searches for neighbors are implemented efficiently and only a small portion of all data points lie in the same area. The latter can be enforced by running LEADERS clustering [VMS04; VP06] in a pre-process and perform DBSCAN on the LEADERS cluster only. The result of this rough DBSCAN [VB09] is not as exact as when performing only DBSCAN, but the overall structures stay the same.

3.1.4 Clustering ensembles

The field of clustering ensembles addresses the combination of information contained in multiple clusterings on the same data. The rational is that objects found to be similar in most clusterings truly belong together. Again, two tasks have to be performed: measuring the similarity of objects and combining them to clusters. The similarity of objects can be represented using a so called co-association matrix counting how often each pair of points falls into the same cluster. Clustering in this matrix can then lead to the final clustering. Fred and Jain [FJ05] create the clustering ensemble by running the k-Means algorithm multiple times to counteract its tendency to terminate in local minima. Under the assumption that data points which are clustered together in most clusterings indeed belong to the same cluster, they create the co-association matrix and apply hierarchical clustering by interpreting it as a distance matrix in order to generate final clusters.

As clustering ensembles are based on common consent on cluster membership, it is important to keep in mind that they are not necessarily superior to classical clustering, if a suitable algorithm is known for the data. In [Kum+18] we created an ensemble of clusterings but did not combine them, as k-Means is already used operationally by the ECMWF [FC11] and provides good results. However, we offer the option to highlight members with similar cluster change characteristics, which corresponds

to one similarity query in clustering ensembles. In contrast, when it is known that a single clustering is only of limited quality, e.g., when it depends on uncertain dimensionality reduction with known inaccuracies as in [KSW19], clustering ensembles are superior to single clusterings. We project a multi-dimensional point set multiple times into 2D using t-SNE, and cluster each projection once with k-Means and a fixed number of clusters. Instead of applying hierarchical clustering on the co-association matrix, we first compute cliques of points with high pair-wise similarity, and combine these greedily to clusters. The rational behind the usage of cliques is that each point has to be similar to multiple others in the cluster, leading to a better separation of clusters than when using standard hierarchical clustering.

3.1.5 Clustering application

Before clustering can be applied, data has to be represented in a way that distance metrics can be applied to it. In order to cluster ensemble members of numerical weather forecasts based on a certain region, each member can be represented as a high-dimensional point, where each grid point lies in its own dimension. Hence, in the case of the operationally used ECMWF clustering, 51 ensemble members lead to 51 high-dimensional data-points, whose dimensionality is reduced by applying principal component analysis and only considering the first dimensions representing at least 80% of the data's variance. k-Means is then run on the projected point set for 2 to 6 clusters and the optimal number of clusters is determined by running a Monte Carlo-based significance test against normally distributed random points in the principal component space.

When clustering parameters are fixed and only few grid-points for the high-dimensional member representation are changed, the clustering results can already change significantly, as we noticed when choosing a slightly different region over Europe than the one defined by the ECMWF. While these changes for k-Means were more pronounced when grid-points with high ensemble variance were added or removed, hierarchical clustering proved to be especially sensitive to variations to the data, while k-Means seems to be more robust and changing gradually in that case; for this reason, k-Means was used in [Kum+18]. The influence of changing variance contained in the data could also be observed when altering the number of principle components used for representing the data.

Density based clustering is impacted a lot by changing distances between data-points, which can be caused by scaling effects or changing embeddings after dimensionality reduction as was the case when applying it on t-SNE projected data.

For future work, it would be worth investigating the stability of fuzzy clustering instead of k-Means for clustering weather forecast ensemble members and adapt the existing visualizations to convey this information. Also allowing one or several outliers in the k-Means clustering by not considering them for the mean computation would be worth investigating. The number of outliers could be determined by the number of members with low co-association values.

3.2 Correlation

Correlation is used to measure how dependent or independent two random variables X and Y are. When a sample (x, y) is drawn from two highly correlated random variables, it is likely that both values are either high or low, whereas negative correlation indicates that a high value of x is likely associated with a low value of y and the other way around. Correlation values lie in a range of $[-1, 1]$, where 1 (-1) indicates perfect positive (negative) correlation.

When multiple data values for the same simulation element, e.g., a grid point, are available, it is often assumed that these values are samples from a random variable following a certain distribution. The correlation between two simulation elements can then be calculated using different methods, such as Pearson's or Spearman's correlation coefficient. In the following, Pearson's correlation coefficient and some properties of it will be briefly explained.

Pearson's correlation coefficient can be calculated by

$$\rho_{X,Y} = \frac{\operatorname{cov}(X, Y)}{\sigma_X \sigma_Y}, \tag{3.1}$$

where $\operatorname{cov}(X, Y)$ is the co-variance between random variables X and Y, and σ_X denotes the standard deviation of X. For a discrete sample of size n, this translates to

$$\begin{aligned} r_{XY} &= \frac{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\underbrace{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}}_{=:s_x}\underbrace{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}}_{=:s_y}} \\ &= \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{(n-1)s_x s_y}. \end{aligned}$$

The mean $(\bar{\cdot})$ and sample standard deviation $(s_\cdot)$ of every simulation element can be cached to accelerate later calculations. It is important to note that this correlation coefficient is not transitive, i.e.,

$$\rho_{XY} > \delta \quad \wedge \quad \rho_{YZ} > \delta \nRightarrow \rho_{XZ} > \delta.$$

However, it can be deduced (see Appendix 8) that

$$\rho_{XZ} \geq \rho_{XY}\rho_{YZ} - \sqrt{1-(\rho_{XY})^2}\sqrt{1-(\rho_{YZ})^2}.$$

This inequality shows that even for simulation elements which are highly correlated to another ele-

ment, one can not assume that these elements are also highly correlated among each other. For this reason, we use a clique clustering algorithm in [Kum+19] to find groups of pair-wise highly correlated elements.

In meteorology, a variation of Pearson's correlation coefficient is used for Ensemble Sensitivity Analysis (ESA). A formal introduction can be found in the articles by Ancell and Hakim [AH07] and Torn and Hakim [TH08]. While Pearson's correlation coefficient is normalized using the standard deviations, ESA scales again with the standard deviation to relate changes in one variable with changes in the other in terms of units instead of standard deviations. Changes in units are relevant as one is more interested in, e.g., by how many degrees the temperature forecast was affected by uncertainties of another variable at another time step.

At this point it is important to note that correlation is not the same as causality. High correlation values can originate from dependencies to unknown variables or even from chance. When many correlation values are computed, it is likely that at least some of them show spurious high correlation.

Confidence tests This is why confidence tests are performed. A correlation value r_0 with a confidence level of 95% means, that under the assumption that the samples are distributed according to an assumed distribution, less than 5% of all correlation values higher than r_0 are spurious. As already the first assumption assuming a specific distribution is often violated (but nevertheless made), and commonly only a small portion of all simulation elements are highly correlated (5% of all simulation elements are suddenly quite a lot), such tests are nevertheless helpful as a sanity check.

When assuming correlation values to be normally distributed, for which the values of the simulation elements have to be normally distributed as well, a Student t-test can be applied to compute the level of significance for each correlation value and a given sample size. It tests against the null hypothesis that the correlation is not significantly different from zero. This test is commonly used because of its simplicity and Edgell et al. [EN84] summarized evidence in 1984 that the t-test is robust against violations of the normality assumption, which makes it applicable in many cases. A thorough introduction into t-tests can be found for example in the book of Hogg et al. [HMC19].

In short, assuming correlation values to be normally distributed with a population mean of 0, the standard error of the mean follows a $\mathcal{X}^2$-distribution. By dividing the sample correlation coefficient by its standard error, the t-statistic

$$t = r\sqrt{\frac{(n-2)}{1-r^2}}$$

is derived, which follows a t-distribution with $n-2$ degrees of freedom. Solving this equation for r, i.e.,

$$r = \sqrt{\frac{t^2}{t^2+n-2}}$$

and plugging in the value at the 95% quantile of this t-distribution, a correlation value is computed which has to be exceeded in order to be significant with a confidence of 95%.

3.3 Dimensionality reduction

Working with high-dimensional data is challenging. With increasing dimensions, more disk space is needed to store the data and data-points become harder to discriminate using euclidean distances. Computer displays can only show information in 2D or at most 3D, which also limits the possibilities for visualizations. Hence, dimensionality reduction is used to transform a data matrix $X \in \mathbb{R}^{n \times p}$, with n observations (e.g., ensemble members) in a p-dimensional space (e.g., each grid point represents a dimension), into $\tilde{X} \in \mathbb{R}^{n \times k}$, where the dimensionality of the data was decreased from p to k, $k < p$. There are many approaches to achieve this. One can simply neglect dimensions (which corresponds to an orthogonal projection), perform a transformation with PCA prior to projection and thereby minimizing the loss of variation in the data, or optimize the projection according to a criteria such as preservation of local neighborhood relations in t-SNE.

3.3.1 Principal component analysis

PCA corresponds to a linear transformation of the original space in a way that the variance of the data-points is maximized along one axis after the other. As these directions correspond to the eigenvectors of decreasingly sorted eigenvalues, the transformation can be computed using a singular value decomposition.

After subtracting the mean of each dimension in X, singular value decomposition decomposes X into

$$X = U \Lambda V^T,$$

with $V^T \in \mathbb{R}^{n \times p}$ representing the transformation of the original coordinate system into the new one using the eigenvectors of X, $U \in \mathbb{R}^{n \times n}$ containing the n data-points in the new coordinate system, and $\Lambda \in \mathbb{R}^{n \times n}$ containing the singular values of X, corresponding to the square root of the absolute value of its eigenvalues. The explained variance eVar_j of each dimension j is calculated by

$$\mathrm{eVar}_j = \frac{\xi_{jj}^2}{\sum_{i=1}^{n} \xi_{ii}^2},$$

where ξ_{ii} is the i^{th} diagonal entry of Λ. To reduce the dimensionality of the data, one can use only the first k dimensions, i.e.,

$$\tilde{X} = \tilde{U} \tilde{\Lambda} \tilde{V}^T,$$

where $\tilde{X} \in \mathbb{R}^{n \times p}$ is the compressed version of X, $\tilde{U} \in \mathbb{R}^{n \times k}$, $\tilde{\Lambda} \in \mathbb{R}^{k \times k}$ and $\tilde{V}^T \in \mathbb{R}^{k \times p}$.

Some key aspects of PCA are that it preserves the maximal variation possible using only the defined number k of dimensions. Further, it is deterministic and can be computed for relatively large data sets. However, while the original dimensions could be interpreted as attributes or grid points in the data, the transformed dimensions represent weighted combinations of them and therefore, the intuitive meaning of each dimension is lost. Nevertheless, the first principal component(s) can often be associated with interpretable properties in the data, such as the different position and strength of a storm in the ensemble [Kel+11], which in that case was the cause for most variance. A detailed discussion of PCA techniques in general is given by Jolliffes [Jol10] and Hannachi [Han04] explains it in the context of meteorology, where it is referred to as Empirical Orthogonal Functions (EOF).

3.3.2 t-distributed Stochastic Neighbor Embedding

t-SNE introduced by Maaten and Hinton [MH08] is a technique well suited for visualizing data, as its projections into 2D are visually pleasing and at the same time preserve many local neighborhood relations of the original high-dimensional data. It is closely related to Stochastic Neighbor Embedding (SNE) [HR03] but leads to better results by using symmetric probabilities between two points and targeting a Student t-distribution for projected points instead of a Gaussian distribution. In the following, a sketch of the algorithm is presented. A detailed description and justification for the different steps can be found in the original reference.

Similarity in the high-dimensional space is defined using conditioned normally distributed probabilities $p_{j|i}$, describing the probability of point x_i choosing x_j as its neighbor. It is computed by

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}.$$

A user defined perplexity value is used to find σ_i in

$$\text{Perp} = 2^{-\sum_j p_{j|i} \log_2 p_{j|i}}.$$

using binary search, where the exponent corresponds to the Shannon entropy. Since $p_{j|i} \neq p_{i|j}$, the joint probability is computed by

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}. \tag{3.2}$$

In the lower-dimensional target space, probabilities are computed using a Student t-distribution to relax an overcrowding problem originating from many high-dimensional points with similar pairwise

distances. The probabilities are computed by

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}}. \tag{3.3}$$

Minimizing the Kullback-Leibler divergence between equation (3.2) and (3.3)

$$KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

on a random initial point configuration in the low dimensional leads to the final projection.

Not all neighborhood relations can be preserved perfectly as only $n + 1$ points can have the same pair-wise distance in n dimensions. This and the fact that gradient descent only converges to local minima leads to varying final projected point configurations, which motivated us to employ clustering ensemble techniques to k-means clusterings on different runs of t-SNE [KSW19].

3.4 The visualization tool Met.3D

Met.3D is being developed under the lead of Marc Rautenhaus [Rau+15a; Rau+15b]. Originally, it was tailored to visualize time-dependent ensemble weather forecasts, but its functionality has grown over the years. By separating the front-end, the rendering (performed by actors on the GPU) and a pipelined data-processing (performed by data sources), data can be requested while rendering stays interactive and the user interface does not freeze. Requests are processed in parallel and intermediate results such as mean fields are cached to reduce hard drive access and re-computations.

Our clustering module in Met.3D [Rau+15a; Rau+15b] uses the library cluster 3.0 [Hoo+04] which offers *C* implementations for k-Means and hierarchical clustering with various distance metrics and linkage criteria. The seeding strategy for k-Means++ was added to investigate its performance as well. For creating clustering ensembles, options to automatically move, shrink or enlarge the clustering domain, or increase the number of clusters or principal components are provided. For iso-contour clustering, signed distance field based clustering is available as well. Clusterings can be loaded and compared as described in [Kum+18]. Diagrams to display the differences in clusters and cluster change characteristics of members proved to be useful for clustering ensembles in general, which is why this part was extended in [KSW19] to perform the clique based clustering ensemble technique presented there. All cluster visualizations can be used for externally generated clusterings as well.

In our correlation implementation into Met.3D, correlation values can be calculated over time or over the ensemble of scalar fields. The physical fields for variable X and Y can be chosen on run-time. The correlation r_{x_i,y_i} can be computed and displayed, where x_i denotes the i^{th} grid point in the field

X (y_i respectively). Alternatively, one vector of values from X is correlated to all grid points of Y. The vector from X can either be the values over the ensemble or over time at a single location, or, summarized over a region per ensemble member using for example the mean. These regions can be selected manually or using correlation cliques (c.f. [Kum+19]). We added connected component labeling as well; components for a negative and positive threshold can be extracted simultaneously and center of mass computations are available per slice or in 3D.

Operations for summarizing multiple correlation fields by computing their mean, maximum or minimum values are selectable. Tracking using optical flow is also hightly customizable, with options to only advect structures, or tracks based on the centers of structures. The recommended setting is summarized in our paper [Kum+19]. Forward and backward tracking can be used independently or combined. Several options are implemented to consider a point being matched. After advection, it either falls withing a structure, is less than one grid point or a user selected multiple of its optical flow away from a structure in the field to match to. Tracking paths can be displayed and arrows as well as the swipe-path can be colored with a transfer function to encode tracking time. These flexible options allow a user to perform different and sophisticated correlation analyses without the need of changing the code, making it applicable for domain experts after only a short introduction to the tool.

3.5 Correcting and inpainting sparse data using neural networks

In this at the time of writing unpublished project we investigate the ability of neural networks to improve renderings of 3D point clouds as recent publications show the potential of neural networks on inpainting tasks. A practical application in meteorology could be the combination and correction of measurement data acquired from LiDAR sensors and dropsondes. This is an ongoing project and many things will be changed for the final solution, which is why this section rather presents the overall approach and experiences gained than highly optimized training scores. As only high level descriptions are given, basic knowledge about deep neural networks is assumed.

The 3D point clouds used here were obtained by the scanning of historical buildings such as castle Neuschwanstein (courtesy of 3D RealityMaps). Each point is attributed with its 3D position (x, y, z), RGB color and surface normals, which were computed in a post-process after the point cloud generation.

With classical approaches, these points can be rendered onto the screen by representing every point as a one pixel splat (see Fig. 3.1 first column). If multiple points fall onto the same pixel, only the one closest to the camera is rendered. Depending on the number of points, the screen resolution and the distance between camera and object, not all pixels are filled and 3D points of occluded structures can become visible.

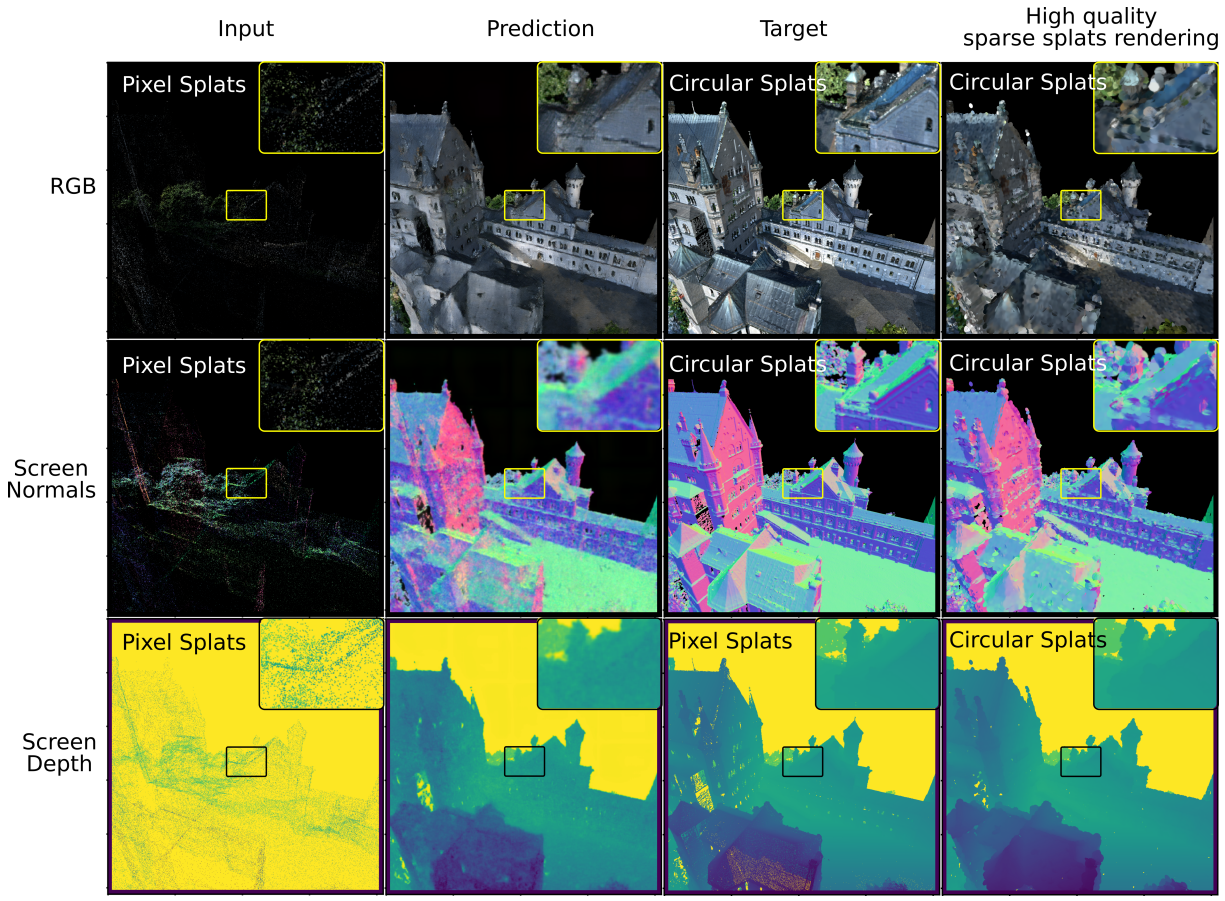

Figure 3.1: Two successive U-Nets applied to a pixel-splats-rendered 3D point cloud. The first U-Net creates a mask to determine which pixels are indeed visible while the second performs the final correction and inpainting. Blended circular splats are used as reference, where the radius was optimized manually. Yellow color in depth channel images encodes absence of points. Inlets show a zoom-in on the center.

By rendering circular splats with a suitable radius in object space instead (see Fig. 3.1 last column), most holes can be avoided. This radius can either be determined experimentally or by the resolution of the scan, which should be uniform everywhere. The splats can now overlap. By rendering them with decreasing opacity from their center and blending splats lying within a certain depth interval, high quality renderings are achieved.

To generate a training data set imitating a less dense point cloud, we randomly remove a large portion of the initial points (approximately 99%) and render the remaining ones using pixel splats. The high-resolution high quality circular splats rendering is used as the ground truth. Even then, some holes remain (see Fig. 3.1, 3rd column), so the target is not perfect. Additionally to color, the depth and screen-space surface normals are stored to provide the network with additional information about the orientation location of points. For rendering, the camera is moved on a sphere around the

whole castle, where the front and side were used for training and the rear side for testing and Fig. 3.1.

When navigating around the object in 3D, the distance between points of structures closer to the camera is wider than for structures far away. Hence, a network architecture with a large perceptional field has to be chosen. Our experiments are based on the U-Net [RFB15] (see Fig. 3.2) since it has been applied successfully for other inpainting tasks [Liu+18]. For early tests on point clouds generated from the ShapeNet data set [Cha+15], where the point distribution was uniform over the whole rendering without big holes, EnhanceNet [SSH17], without its last upsampling layer, lead to superior results. Since it is based on 10 successive residual blocks, it can only fill holes if the points are not much further than 10 pixels apart, so it failed for the more challenging Neuschwanstein data set.

The U-Net was able to reliably fill holes and reconstruct the outlines of the building structures for Neuschwanstein. However, we noticed some spilling of pixels of occluded structures, for example the green color of trees from the inner courtyard was spilled onto the walls. Inspired by the network structure of Super SlowMo [Jia+18], we combined two U-Nets consecutively, where the first's task was to only predict a mask predicting whether a rendered pixel is visible in the ground truth or not. This mask is then passed as an additional channel to the second U-Net, so the whole network could still be trained end-to-end. More precisely, an additional sigmoid layer was added after the first U-Net to push values towards 0 or 1. Additionally, the learned mask is then masked to 0 where no pixels are rendered in the input image. Both U-Nets use $C = 32$ channels, a depth of $i = 6$, 3x3 2D convolutional kernels except for the first block (5x5) and only one convolution per layer (see Fig. 3.2 without the red blocks).

As input we use three channels for RGB, three for screen normals and one for depth, all obtained by the pixel splats rendering of the thinned-out point cloud. As target, circular splats rendering was used for RGB (0.5×L1-loss + perceptual-loss + texture-loss) and screen normals (cosine-loss + 0.5×L1-loss), but pixel splats for the depth (L1-loss), since the correct depth values are needed to learn a good depth mask (L1-loss + binary-cross-entropy-loss). For perceptual loss, kernels conv_1 to conv_16 of VGG-19 [SZ15] with weights normalizing the effect of the channels were used. Texture loss [GEB16] was used with kernels conv_1, conv_3 and conv_5.

The results shown in Fig. 3.1 were achieved using a batch size of 6 with a resolution of 256x256 (random non-empty patches from the rendered 1024x1024 images), Adam optimizer ($\beta = (0.5, 0.999)$) with an initial learning rate of 1e-3, a decay of factor 0.8 every 50 epochs and 720 epochs of training on 103 batches. For evaluating the loss, a border of 16 pixels around the patches was removed.

The network is able to inpaint missing parts while retaining sharp edges of the structures. The normals as well as the depth are a bit blurry, but the RGB prediction contains more details than the classical high quality splats rendering.

Sangram Gupta experimented in his Master's thesis (*Inpainting sparse Point Cloud Renderings using Convolutional Neural Networks*) with replacing the standard convolutional kernels by partial convolutions [Liu+18], to avoid spilling of invalid pixels, and deformable convolutions presented by Dai et al. [Dai+17] and Zhu et al. [Zhu+19] to account for the non-regularity of point samples. Exploring this further and also changing the distance of the camera to the scene would be worth investigating. Also a lighting correction could be applied to the point clouds. Since the scanning process takes some time, lighting conditions change, e.g., due to moving clouds. Correcting the lighting on the fly while rendering as proposed by Kanzok et al. [KLR12] could therefore further improve the training process.

Once the best combination of architecture and kernels, loss-functions and data representation is known, the insights gained in this project can be used to complete and correct measurement data as already mentioned. For this purpose, new training data has to be generated to train the model on it. Another possible application could be the rendering of particle simulations.

3.6 Slice interpolation using neural networks

The second neural-network-based project addresses the problem of interpolating intermediate levels or time steps based on enclosing ones. Most work published addresses this issue for videos, where optical flow based approaches seem to lead to the best results.

More precisely, we explore the method by Niklaus and Liu [NL18]. After estimating the optical flow with the PWC-Net [Sun+18] between the enclosing frames, they warp both images as well as context images to the intermediate time-step. These eight channels are then fed into a GridNet predicting the final image. As we had troubles achieving a stable training of the GridNet, we replaced it by a U-Net.

Setting The training data set is generated from a COSMO ensemble simulation over Germany, where we create non-overlapping patches from the temperature field. The patch size is 300×300 grid-points and we randomly skip between 0 to 3 vertical levels between the lower (input level 1), middle (target) and upper slice (input level 2). The differing distance between levels is ignored in this setting to avoid interpolated values for the training data.

The optical flow between the enclosing levels is then estimated using PWC-Net [Sun+18]. Therefore, the values of both input layers are standardized to follow a normal distribution with mean 0 and standard deviation 1 each. This step is necessary since the mean temperature as well as the variability of it changes significantly across levels. Thereafter, both input layers are normalized into a range of $[0, 1]$ using the minimum and maximum value over both fields as PWC-Net is only trained for values in the range $[0, 1]$. After warping, all channels are denormalized again and the warped ones are rescaled to have linearly interpolated mean and standard deviation of the enclosing levels. For training, all these levels are then normalized with the same values to make different patches comparable.

Additionally, a context map is generated by the response of the first convolutional layer of ResNet-18 [He+16], with adapted stride 1 to preserve the resolution. These are warped with the same flow as the original levels and again, all four context maps are normalized simultaneously for training. Data augmentation is performed by training on random crops of size 256×256, flipped x and y coordinates and the order of the enclosing levels.

For the example shown in Fig. 3.3, a U-Net including the red operations in Fig. 3.2 and an additional 1x1 2D convolution directly after the input is used, with $C = 10$ channels, $i = 4$ downsampling levels, a batch size of 12, standard L1-loss, Adam optimizer ($\beta = (0.5, 0.999)$) with learning rate of 2e-4 with a decay of factor 0.7 every 20 epochs and 320 epochs of training. The EnhanceNet shown was trained with $C = 15$ channels, batch size of 25 and the same loss, optimizer and learning rate. Using more channels did not improve the networks further. For training, 18000 patches are used.

Evaluated over 1995 validation files, the mean error reduction compared to linear interpolation was 46.7% with a standard deviation of 54.0. Compared to the warped layers, the error was reduced by 39.8% (33.9%) when warping the lower (upper) level. The discrepancy is probably caused by the richer details in levels closer to the surface. In inference mode, the interpolation took roughly 2.3 milliseconds on a NVIDIA GTX 1070.

The EnhanceNet in Fig. 3.3 performed slightly worse, the mean error reduction compared to linear interpolation was 41.8% but with a lower standard deviation of 51.3. The mean error reduction compared to the warped levels was 30.9% and 22.5%. Inference took 3.3 milliseconds.

These preliminary results show that neural networks indeed can help to improve interpolation for scientific data. As some single cases also showed worse results than linear interpolation, a criteria to ensure a certain quality of interpolation should be included in future work.

Sukanya Raju explored in her Master's thesis (*Scientific Data Interpolation Using Convolutional Neural Networks*) the capabilities of the network architecture used for Super SlowMo [Jia+18], where in a first U-Net a rough estimation of the optical flow is generated while the second one improves it. Additionally, visibility maps are generated to account for occlusions of pixels in one of the two frames. The intermediate frames are then generated by warping enclosing frames. This approach allows to interpolate an arbitrary number of intermediate frames and leads to slightly better results than the direct U-Net-based prediction and would therefore be worth to be investigated in more detail.

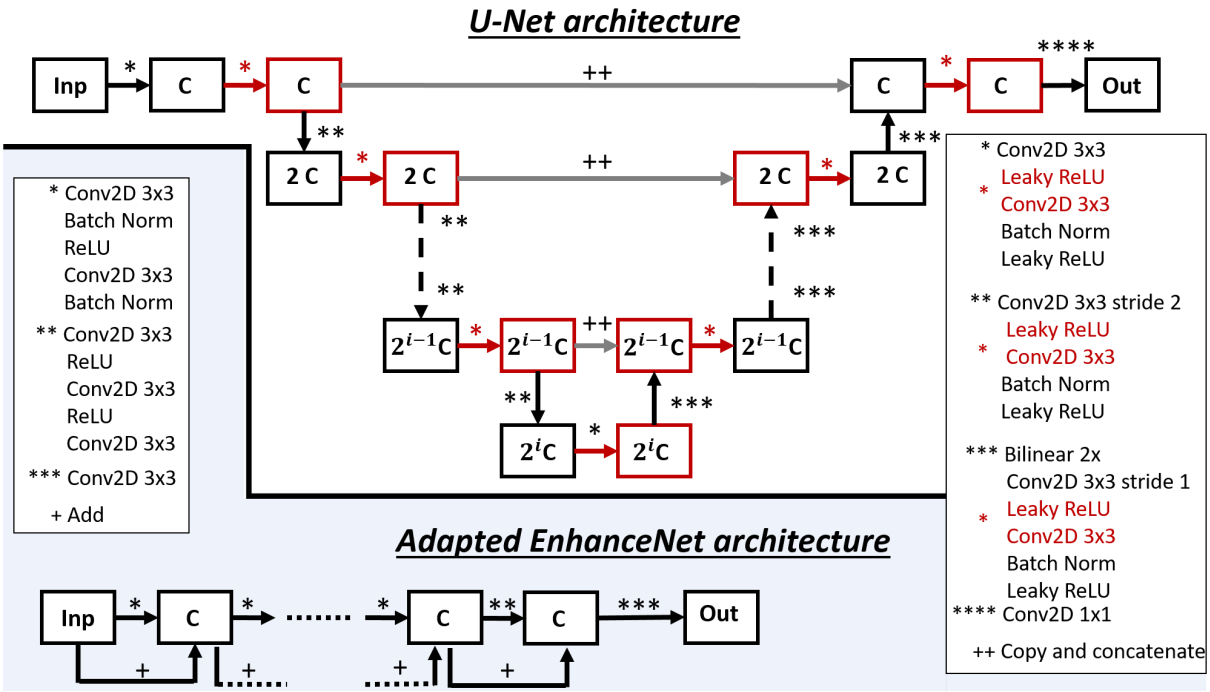

Figure 3.2: Neural network architectures used. Operations are performed on the arrows, while the red operations were optional. The number of channels is indicated in each box, where *Inp*(*Out*) refers to the number of input (output) channels and the number of channels *C* in the network has to be set. Our adapted EnhanceNet does not perform upsampling in the final layer.

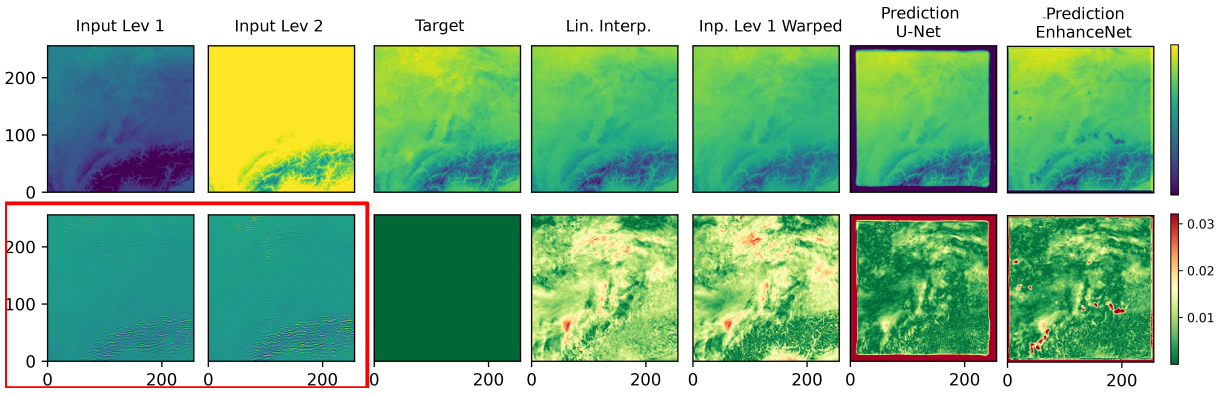

Figure 3.3: The U-Net and EnhanceNet applied to interpolate the intermediate level of a scalar temperature field. 7 levels were skipped between input level 1 and input level 2. Shown is a region over southern Germany including parts of the Alps. The first row shows the temperature while the second shows the two context maps (in red rectangle) and the absolute difference to the target. The mean error compared to linear interpolation was reduced by 74.3% by the U-Net (67.8% EnhanceNet) and compared to the warped input level 1 by 59.8% (49.6% EnhanceNet). A border of 16 pixels is excluded from the loss calculation.

4

Paper A: Visualizing Confidence in Cluster-based Ensemble Weather Forecast Analyses[1]

Abstract of paper In meteorology, cluster analysis is frequently used to determine representative trends in ensemble weather predictions in a selected spatio-temporal region, e.g., to reduce a set of ensemble members to simplify and improve their analysis. Identified clusters (i.e., groups of similar members), however, can be very sensitive to small changes of the selected region, so that clustering results can be misleading and bias subsequent analyses. In this article, we –a team of visualization scientists and meteorologists– deliver visual analytics solutions to analyze the sensitivity of clustering results with respect to changes of a selected region. We propose an interactive visual interface that enables simultaneous visualization of a) the variation in composition of identified clusters (i.e., their robustness), b) the variability in cluster membership for individual ensemble members, and c) the uncertainty in the spatial locations of identified trends. We demonstrate that our solution shows meteorologists how representative a clustering result is, and with respect to which changes in the selected region it becomes unstable. Furthermore, our solution helps to identify those ensemble members which stably belong to a given cluster and can thus be considered similar. In a real-world application case we show how our approach is used to analyze the clustering behavior of different regions in a forecast of "Tropical Cyclone Karl", guiding the user towards the cluster robustness information required for subsequent ensemble analysis.

Contribution The method development and implementation was done by the first author. The plain version of contour probability plots was developed by Bianca Tost and Marc Rautenhaus, insights of the case study were contributed by Marlene Baumgart and Michael Riemer. Parts of the background

infrastructure was also implemented by Bianca Tost. Discussions with Rüdiger Westermann, Marc Rautenhaus and the other co-authors led to the final paper.

5

Paper B: Visual Analysis of the Temporal Evolution of Ensemble Forecast Sensitivities[2]

Abstract of paper Ensemble sensitivity analysis (ESA) has been established in the atmospheric sciences as a correlation-based approach to determine the sensitivity of a scalar forecast quantity computed by a numerical weather prediction model to changes in another model variable at a different model state. Its applications include determining the origin of forecast errors and placing targeted observations to improve future forecasts. We—a team of visualization scientists and meteorologists—present a visual analysis framework to improve upon current practice of ESA. We support the user in selecting regions to compute a meaningful target forecast quantity by embedding correlation-based grid-point clustering to obtain statistically coherent regions. The evolution of sensitivity features computed via ESA are then traced through time, by integrating a quantitative measure of feature matching into optical-flow-based feature assignment, and displayed by means of a swipe-path showing the geo-spatial evolution of the sensitivities. Visualization of the internal correlation structure of computed features guides the user towards those features robustly predicting a certain weather event. We demonstrate the use of our method by application to real-world 2D and 3D cases that occurred during the 2016 NAWDEX field campaign, showing the interactive generation of hypothesis chains to explore how atmospheric processes sensitive to each other are interrelated.

Contribution The method development and implementation was done by the first author, while the evaluation was done by Marc Rautenhaus and Michael Riemer. Discussions with the co-authors led to the final paper.

[2]

6

Paper C: Cluster-based Analysis of Multi-Parameter Distributions in Cloud Simulation Ensembles[3]

Abstract of paper The proposed approach enables a comparative visual exploration of multi-parameter distributions in time-varying 3D ensemble simulations. To investigate whether dominant trends in such distributions occur, we consider the simulation elements in each dataset—per ensemble member and time step—as elements in the multi-dimensional parameter space, and use t-SNE to project these elements into 2D space. To find groups of elements with similar parameter values in each time step, the resulting projections are clustered via k-Means. Since elements with similar data values can be disconnected in one single projection, we compute an ensemble of projections using multiple t-SNE runs and use evidence accumulation to determine sets of elements that are stably clustered together. We build upon per-cluster multi-parameter distribution functions to quantify cluster similarity, and merge clusters in different ensemble members. By applying the proposed approach to a time-varying ensemble, the temporal development of clusters, and in particular their stability over time can be analyzed. We apply this approach to analyze a time-varying ensemble of 3D cloud simulations. The visualizations via t-SNE, parallel coordinate plots and scatter plot matrices show dependencies between the simulation results and the simulation parameters used to generate the ensemble, and they provide insight into the temporal ensemble variability regarding the major trends in the multi-parameter distributions.

Contribution The method development, their implementation and application were done by the first author. Josef Stumpfegger implemented the Parallel Coordinate tool. Discussions with Rüdiger Westermann led to the final paper.

[3]This article was published in Vision, Modeling and Visualization, Alexander Kumpf, Josef Stumpfegger, Rüdiger Westermann, Cluster-based Analysis of Multi-Parameter Distributions in Cloud Simulation Ensembles

7 Conclusion

In the work associated with this thesis we investigate how information retrieved from ensemble data can be used and be visualized to gain a deeper understanding of the data presented. Starting from exploring sensitivities when clustering ensemble members, we continue using the ensemble data to reveal sensitivities between locations, time steps and variables, and finally provide means to relate differences in whole simulations to their initial simulation parameters. Early results of current projects demonstrate the usefulness of deep neural networks for interpolating and completing scientific data.

We provide means to efficiently find a suitable clustering when clustering ensemble members and at the same time provide information on its stability. We hope that our work motivates authors to justify and discuss the choice of the explicit clustering used in their work. Further, experiments performed for the project presented in [RKW19] show the potential of the abstract member and cluster visualizations to investigate ensembles of 2D points in general.

The interactive correlation method in Met.3D improves upon common practice of ensemble sensitivity analysis by providing reasonable region selections with automated tracking and overview plots. The modular realization in Met.3D makes the methods applicable for other correlation analyses as well, utilizing caching mechanisms to optimize performance.

The identification of structures by their multi-parameter distributions allows for a comparison of data sets or parts of data sets independent of their exact spatial size and shape to draw conclusions about relations to initial parameters. In an ongoing project, we extend this by combining multiple linked views to select parameter ranges and compare ensemble members based on their value distributions.

Meteorology provides a rich pool of visualization problems which are yet to solve. Working together with domain experts is crucial as they develop a good understanding of the data over the years which can be incorporated into the methods. It seems that in many domains the current state-of-the-art

visualization methods are not used yet in their full extend which makes their application and adaption to the specific problems worthwhile. As each specific case provides obstacles on its own, also the development of new techniques can be inspired by this. In this context, the project Waves to Weather was the perfect opportunity to get in contact with the experts, discuss their problems and aims and while gaining some understanding of their field through numerous meetings and presentations, collaborations and the combination of at first seemingly unrelated methods lead to major advancements in understanding their data.

8

Appendix - Dependency between two Pearson's correlation coefficients

In the following we will briefly derive the relationship between Pearson's correlation coefficients between pairs of three points X, Y and Z. Assuming that the sample mean of each variable has already been subtracted, Pearson's correlations coefficient computation reduces to calculating the cosine of the angle between x and y, i.e.,

$$r_{x,y} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} = \frac{\langle x, y\rangle}{\sqrt{\langle x, x\rangle}\sqrt{\langle y, y\rangle}},$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product. Since $\langle x, y\rangle = \|x\|\|y\| \cos \measuredangle(x, y)$, it follows that

$$r_{x,y} = \frac{\cos \measuredangle(x, y)}{\sqrt{\cos \measuredangle(x, x)}\sqrt{\cos \measuredangle(y, y)}} = \cos \measuredangle(x, y).$$

For correlation values $r_{x,y}$ and $r_{y,z}$ close to one, each angle is below 90°. Since $\cos(\xi)$ is monotonically decreasing for $\xi \in [0°, 180°]$, for $\alpha := \measuredangle(x, y)$, $\beta := \measuredangle(y, z)$ and $\gamma := \measuredangle(x, z)$,

$$\gamma < \alpha + \beta \quad \Rightarrow \quad \cos(\gamma) > \cos(\alpha + \beta).$$

Using $\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$ and replacing $\sin(\cdot) = \sqrt{1 - \cos^2(\cdot)}$ and cosines by correlation coefficients, this finally leads to

$$r_{x,z} \geq r_{x,y} r_{y,z} - \sqrt{1 - (r_{x,y})^2}\sqrt{1 - (r_{y,z})^2}.$$

Bibliography

[1] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. “Visual methods for analyzing time-oriented data”. In: *IEEE Transaction on Visualization and Computer Graphics* 14.1 (2007), pp. 47–60 (cit. on p. 9).

[2] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. “Visualizing time-oriented data–a systematic view”. In: *Computers & Graphics* 31.3 (2007), pp. 401–409 (cit. on p. 9).

[3] B. Ancell and G. J. Hakim. “Comparing Adjoint- and Ensemble-Sensitivity Analysis with Applications to Observation Targeting”. In: *Monthly Weather Review* 135.12 (2007), pp. 4117–4134 (cit. on pp. 2, 24).

[4] A. Anil Valsangkar, J. Merwin Monteiro, V. Narayanan, I. Hotz, and V. Natarajan. “An Exploratory Framework for Cyclone Identification and Tracking”. In: *IEEE Transactions on Visualization and Computer Graphics* (Feb. 2018), pp. 1–1 (cit. on p. 16).

[5] A. Antonov, G. Lohmann, M. Ionita, M. Dima, and L. Linsen. “An interactive visual analysis tool for investigating teleconnections in climate simulations”. In: *Environmental Earth Sciences* 78.10 (2019), p. 294 (cit. on p. 14).

[6] D. Arthur and S. Vassilvitskii. “k-means++: The advantages of careful seeding”. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics. 2007, pp. 1027–1035 (cit. on pp. 11, 20).

[7] T. Athawale, E. Sakhaee, and A. Entezari. “Isosurface visualization of data with nonparametric models for uncertainty”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2015), pp. 777–786 (cit. on p. 13).

[8] N. Bansal, A. Blum, and S. Chawla. “Correlation clustering”. In: *Machine Learning* 56 (2004), pp. 89–113 (cit. on p. 14).

[9] M. Baumgart, M. Riemer, V. Wirth, F. Teubler, and S. T. Lang. “Potential vorticity dynamics of forecast errors: A quantitative case study”. In: *Monthly Weather Review* 146.5 (2018), pp. 1405–1425 (cit. on p. 10).

[10] G.-P. Bonneau, H.-C. Hege, C. R. Johnson, M. M. Oliveira, K. Potter, P. Rheingans, and T. Schultz. “Overview and state-of-the-art of uncertainty visualization”. In: *Scientific Visualization*. Springer, 2014, pp. 3–27 (cit. on p. 8).

[11] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen. “Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications.” In: *Eurographics (STARs)*. 2013, pp. 39–63 (cit. on p. 8).

[12] G. Bradski and A. Kaehler. “OpenCV”. In: *Dr. Dobb’s journal of software tools* 3 (2000) (cit. on p. 16).

[13] P. T. Bremer, G. Weber, J. Tierny, V. Pascucci, M. Day, and J. Bell. “Interactive Exploration and Analysis of Large-Scale Simulations Using Topology-Based Data Segmentation”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.9 (Sept. 2011), pp. 1307–1324 (cit. on p. 15).

[14] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. “Graphical methods for data analysis. Wadsworth & Brooks”. In: *Cole Statistics/Probability Series* (1983) (cit. on p. 13).

[15] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. “Shapenet: An information-rich 3d model repository”. In: *arXiv preprint arXiv:1512.03012* (2015) (cit. on p. 30).

[16] C.-K. Chen, C. Wang, K.-L. Ma, and A. T. Wittenberg. “Static correlation visualization for large time-varying volume data”. In: *Visualization Symposium (PacificVis), 2011 IEEE Pacific*. IEEE. 2011, pp. 27–34 (cit. on p. 14).

[17] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. “Deformable convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 764–773 (cit. on p. 31).

[18] E. D. Demaine and N. Immorlica. “Correlation clustering with partial information”. In: *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*. Springer, 2003, pp. 1–13 (cit. on p. 14).

[19] I. Demir, C. Dick, and R. Westermann. “Multi-Charts for Comparative 3D Ensemble Visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014) (cit. on p. 13).

[20] I. Demir, J. Kehrer, and R. Westermann. “Screen-space Silhouettes for Visualizing Ensembles of 3D Isosurfaces”. In: *Proc. IEEE Pacific Visualization Symp. (Visualization Notes)*. 2016 (cit. on p. 8).

[21] H. Doraiswamy, V. Natarajan, and R. S. Nanjundiah. “An Exploration Framework to Identify and Track Movement of Cloud Systems”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2896–2905 (cit. on p. 16).

[22] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. “Flownet: Learning optical flow with convolutional networks”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2758–2766 (cit. on p. 16).

[23] S. Dutta and H.-W. Shen. “Distribution driven extraction and tracking of features for time-varying data analysis”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2015), pp. 837–846 (cit. on p. 13).

[24] S. E. Edgell and S. M. Noon. “Effect of violation of normality on the t test of the correlation coefficient.” In: *Psychological bulletin* 95.3 (1984), p. 576 (cit. on p. 24).

[25] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” In: *Kdd*. Vol. 96. 34. 1996, pp. 226–231 (cit. on pp. 12, 21).

[26] B. S. Everitt, S. Landau, M. Leese, and D. Stahl. “Cluster Analysis”. In: *Cluster Analysis*. John Wiley & Sons, Ltd, 2011 (cit. on p. 19).

[27] Z. Fang, T. Möller, G. Hamarneh, and A. Celler. “Visualization and Exploration of Time-varying Medical Image Data Sets”. In: *Proceedings of Graphics Interface 2007*. 2007, pp. 281–288 (cit. on p. 14).

[28] G. Farnebäck. “Two-frame motion estimation based on polynomial expansion”. In: *Scandinavian conference on Image analysis*. Springer. 2003, pp. 363–370 (cit. on p. 16).

[29] L. Ferranti and S. Corti. “New clustering products”. In: *ECMWF Newsl* 127 (2011), pp. 6–11 (cit. on pp. 1, 11, 21).

[30] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. “Visual Analysis of Spatial Variability and Global Correlations in Ensembles of Iso-Contours”. In: *Computer Graphics Forum* 35.3 (2016), pp. 221–230 (cit. on pp. 10, 11).

[31] F. Ferstl. “Visualizing the Variability in Ensemble Simulations”. Dissertation. München: Technische Universität München, 2016 (cit. on p. 10).

[32] F. Ferstl, K. Bürger, and R. Westermann. “Streamline Variability Plots for Characterizing the Uncertainty in Vector Field Ensembles”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 767–776 (cit. on pp. 10, 11).

[33] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. “Time-hierarchical Clustering and Visualization of Weather Forecast Ensembles”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 831–840 (cit. on pp. 9, 11).

[34] A. Fofonov, V. Molchanov, and L. Linsen. “Visual Analysis of Multi-Run Spatio-Temporal Simulations Using Isocontour Similarity for Projected Views”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.8 (Aug. 2016), pp. 2037–2050 (cit. on p. 9).

[35] A. Fofonov and L. Linsen. “MultiVisA: Visual Analysis of Multi-run Physical Simulation Data using Interactive Aggregated Plots.” In: *VISIGRAPP (3: IVAPP)*. 2018, pp. 62–73 (cit. on p. 9).

[36] D. Fourure, R. Emonet, É. Fromont, D. Muselet, A. Trémeau, and C. Wolf. “Residual conv-deconv grid network for semantic segmentation”. In: *arXiv preprint arXiv:1707.07958* (2017) (cit. on p. 17).

[37] T. H. A. Frame, M. H. P. Ambaum, S. L. Gray, and J. Methven. “Ensemble prediction of transitions of the North Atlantic eddy-driven jet”. In: *Quarterly Journal of the Royal Meteorological Society* 137.658 (2011), pp. 1288–1297 (cit. on p. 11).

[38] A. L. N. Fred and A. K. Jain. “Combining multiple clusterings using evidence accumulation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.6 (June 2005), pp. 835–850 (cit. on pp. 12, 21).

[39] L. A. Gatys, A. S. Ecker, and M. Bethge. “Image style transfer using convolutional neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423 (cit. on p. 30).

[40] H. Griethe, H. Schumann, et al. “The visualization of uncertain data: Methods and problems.” In: *SimVis*. 2006, pp. 143–156 (cit. on p. 7).

[41] A. Hannachi. “A primer for EOF analysis of climate data”. In: *Department of Meteorology, University of Reading* (2004), pp. 1–33 (cit. on p. 26).

[42] S. Havre, B. Hetzler, and L. Nowell. “ThemeRiver: Visualizing theme changes over time”. In: *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*. IEEE. 2000, pp. 115–123 (cit. on p. 8).

[43] S. Hazarika, S. Dutta, H. Shen, and J. Chen. “CoDDA: A Flexible Copula-based Distribution Driven Analysis Framework for Large-Scale Multivariate Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 1214–1224 (cit. on p. 13).

[44] S. Hazarika, A. Biswas, and H.-W. Shen. “Uncertainty visualization using copula-based analysis in mixed distribution models”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017), pp. 934–943 (cit. on p. 13).

[45] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 16, 32).

[46] G. E. Hinton and S. T. Roweis. “Stochastic neighbor embedding”. In: *Advances in neural information processing systems*. 2003, pp. 857–864 (cit. on p. 26).

[47] J. L. Hintze and R. D. Nelson. “Violin Plots: A Box Plot-Density Trace Synergism”. In: *The American Statistician* 52.2 (1998), pp. 181–184 (cit. on p. 13).

[48] R. V. Hogg, J. McKean, and A. T. Craig. *Introduction to Mathematical Statistics, Global Edition*. Pearson Education Canada, 2019 (cit. on p. 24).

[49] M. J. de Hoon, S. Imoto, J. Nolan, and S. Miyano. “Open source clustering software”. In: *Bioinformatics* 20.9 (2004), pp. 1453–1454 (cit. on p. 27).

[50] B. K. Horn and B. G. Schunck. “Determining optical flow”. In: *Artificial Intelligence* 17.1 (1981), pp. 185–203 (cit. on p. 16).

[51] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. “Flownet 2.0: Evolution of optical flow estimation with deep networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2462–2470 (cit. on p. 16).

[52] A. Inselberg. “The plane with parallel coordinates”. In: *The visual computer* 1.2 (1985), pp. 69–91 (cit. on p. 8).

[53] M. Jarema, J. Kehrer, and R. Westermann. “Comparative visual analysis of transport variability in flow ensembles”. In: (2016) (cit. on p. 9).

[54] M. Jarema, I. Demir, J. Kehrer, and R. Westermann. “Comparative visual analysis of vector field ensembles”. In: *IEEE Conference on Visual Analytics Science and Technology (VAST)*. 2015, pp. 81–88 (cit. on p. 13).

[55] D. Jen, P. Parente, J. Robbins, C. Weigle, R. Taylor II, A. Burette, and R. Weinberg. “Imagesurfer: A tool for visualizing correlations between two volume scalar fields”. In: *Proceedings of IEEE Visualization*. 2004, pp. 529–536 (cit. on p. 14).

[56] G. Ji and H.-W. Shen. “Feature tracking using earth mover’s distance and global optimization”. In: *Pacific Graphics*. Vol. 2. 2006 (cit. on p. 15).

[57] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz. “Super slomo: High quality estimation of multiple intermediate frames for video interpolation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9000–9008 (cit. on pp. 17, 30, 32).

[58] C. R. Johnson and A. R. Sanderson. “A next step: Visualizing errors and uncertainty”. In: *IEEE Computer Graphics and Applications* 23.5 (2003), pp. 6–10 (cit. on p. 7).

[59] I. Jolliffe. *Principal Component Analysis*. 2nd ed. Springer Series in Statistics. Springer, 2010 (cit. on pp. 12, 26).

[60] T. Kanzok, L. Linsen, and P. Rosenthal. “On-the-fly luminance correction for rendering of inconsistently lit point clouds”. In: (2012) (cit. on p. 31).

[61] J. Kehrer and H. Hauser. “Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.3 (Mar. 2013), pp. 495–513 (cit. on p. 9).

[62] J. H. Keller, S. C. Jones, J. L. Evans, and P. A. Harr. “Characteristics of the TIGGE multimodel ensemble prediction system in representing forecast variability associated with extratropical transition”. In: *Geophysical Research Letters* 38.12 (June 2011), pp. L12802+ (cit. on p. 26).

[63] M. Kern, T. Hewson, A. Schäfler, R. Westermann, and M. Rautenhaus. “Interactive 3D Visual Analysis of Atmospheric Fronts”. In: *IEEE Transactions on Visualization and Computer Graphics* (2018), pp. 1–1 (cit. on p. 9).

[64] M. Kern, T. Hewson, F. Sadlo, R. Westermann, and M. Rautenhaus. “Robust Detection and Visualization of Jet-stream Core Lines in Atmospheric Flow”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 893–902 (cit. on p. 9).

[65] M. Kern and R. Westermann. “Clustering Ensembles of 3D Jet-Stream Core Lines”. In: *Vision, Modeling and Visualization*. The Eurographics Association, 2019 (cit. on p. 11).

[66] J. B. Kruskal and M. Wish. *Multidimensional scaling*. Vol. 11. Sage, 1978 (cit. on p. 12).

[67] A. Kumpf, M. Rautenhaus, M. Riemer, and R. Westermann. “Visual Analysis of the Temporal Evolution of Ensemble Forecast Sensitivities”. In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 98–108 (cit. on pp. 4, 5, 9, 14–16, 24, 28).

[68] A. Kumpf, J. Stumpfegger, and R. Westermann. “Cluster-based Analysis of Multi-Parameter Distributions in Cloud Simulation Ensembles”. In: *Vision, Modeling and Visualization*. The Eurographics Association, 2019 (cit. on pp. 4, 5, 12, 13, 22, 27).

[69] A. Kumpf, B. Tost, M. Baumgart, M. Riemer, R. Westermann, and M. Rautenhaus. “Visualizing Confidence in Cluster-based Ensemble Weather Forecast Analyses”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pp. 109–119 (cit. on pp. 4, 5, 9–11, 13, 20–22, 27).

[70] S. Leistikow, K. Huesmann, A. Fofonov, and L. Linsen. “Aggregated Ensemble Views for Deep Water Asteroid Impacts Simulations”. In: *IEEE Computer Graphics and Applications* (2019) (cit. on p. 9).

[71] H. Li, C.-W. Fu, Y. Li, and A. Hanson. “Visualizing large-scale uncertainty in astrophysical data”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1640–1647 (cit. on p. 8).

[72] T. Liebmann, G. H. Weber, and G. Scheuermann. “Hierarchical Correlation Clustering in Multiple 2D Scalar Fields”. In: *Computer Graphics Forum* (2018) (cit. on p. 14).

[73] J. Lin. “Divergence measures based on the Shannon entropy”. In: *IEEE Transactions on Information theory* 37.1 (1991), pp. 145–151 (cit. on p. 14).

[74] L. Linsen. *Point cloud representation*. Univ., Fak. für Informatik, Bibliothek Technical Report, Faculty of Computer, 2001 (cit. on p. 17).

[75] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. “Image inpainting for irregular holes using partial convolutions”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 85–100 (cit. on pp. 17, 30, 31).

[76] S. Liu, J. A. Levine, P.-T. Bremer, and V. Pascucci. “Gaussian mixture model based volume visualization”. In: *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*. IEEE. 2012, pp. 73–77 (cit. on p. 13).

[77] S. Lloyd. “Least squares quantization in PCM”. In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137 (cit. on pp. 11, 19).

[78] A. L. Love, A. Pang, and D. L. Kao. “Visualizing spatial multivalue data”. In: *IEEE Computer Graphics and Applications* 25.3 (May 2005), pp. 69–79 (cit. on pp. 8, 13).

[79] B. D. Lucas, T. Kanade, et al. “An iterative image registration technique with an application to stereo vision.” In: *IJCAI*. Vol. 81. 1981, pp. 674–679 (cit. on p. 16).

[80] L. v. d. Maaten and G. Hinton. “Visualizing data using t-SNE”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605 (cit. on pp. 12, 26).

[81] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. “The planar k-means problem is NP-hard”. In: *International Workshop on Algorithms and Computation*. Springer. 2009, pp. 274–285 (cit. on p. 19).

[82] M. Mirzargar, R. T. Whitaker, and R. M. Kirby. “Curve boxplot: Generalization of boxplot for ensembles of curves”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2654–2663 (cit. on p. 10).

[83] V. Molchanov and L. Linsen. “Interactive design of multidimensional data projection layout”. In: (2014) (cit. on p. 12).

[84] V. Molchanov and L. Linsen. “Overcoming the Curse of Dimensionality When Clustering Multivariate Volume Data.” In: *VISIGRAPP (3: IVAPP)*. 2018, pp. 29–39 (cit. on p. 12).

[85] V. Molchanov and L. Linsen. “Upsampling for Improved Multidimensional Attribute Space Clustering of Multifield Data”. In: *Information* 9.7 (2018), p. 156 (cit. on p. 12).

[86] C. Muelder and K. L. Ma. “Interactive feature extraction and tracking by utilizing region coherency”. In: *2009 IEEE Pacific Visualization Symposium*. Apr. 2009, pp. 17–24 (cit. on p. 16).

[87] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi. “Edgeconnect: Generative image inpainting with adversarial edge learning”. In: *arXiv preprint arXiv:1901.00212* (2019) (cit. on p. 17).

[88] S. Niklaus and F. Liu. “Context-aware synthesis for video frame interpolation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1701–1710 (cit. on pp. 17, 31).

[89] S. Niklaus, L. Mai, and F. Liu. “Video Frame Interpolation via Adaptive Separable Convolution”. In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017 (cit. on p. 17).

[90] T. Nocke, H. Schumann, and U. Böhm. “Methods for the visualization of clustered climate data”. In: *Computational Statistics* 19.1 (2004), pp. 75–94 (cit. on p. 8).

[91] H. Obermaier and K. Joy. “Future challenges for ensemble visualization”. In: *Computer Graphics and Applications, IEEE* 34.3 (May 2014), pp. 8–11 (cit. on p. 8).

[92] S. Oeltze, D. J. Lehmann, A. Kuhn, G. Janiga, H. Theisel, and B. Preim. “Blood flow clustering and applications in virtual stenting of intracranial aneurysms”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.5 (2014), pp. 686–701 (cit. on p. 20).

[93] O. Pele and M. Werman. “Fast and robust earth mover’s distances”. In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE. 2009, pp. 460–467 (cit. on p. 15).

[94] O. Pele and M. Werman. “The quadratic-chi histogram distance family”. In: *European conference on computer vision*. Springer. 2010, pp. 749–762 (cit. on p. 15).

[95] T. Pfaffelmoser, M. Mihai, and R. Westermann. “Visualizing the Variability of Gradients in Uncertain 2D Scalar Fields.” In: *IEEE Transactions on Visualization and Computer Graphics* (2013) (cit. on p. 8).

[96] T. Pfaffelmoser and R. Westermann. “Visualization of Global Correlation Structures in Uncertain 2D Scalar Fields”. In: *Computer Graphics Forum*. Vol. 31. Wiley Online Library. 2012, pp. 1025–1034 (cit. on p. 14).

[97] T. Pfaffelmoser and R. Westermann. “Visualizing Contour Distributions in 2D Ensemble Data”. In: *EuroVis-Short Papers*. The Eurographics Association. 2013, pp. 55–59 (cit. on p. 10).

[98] K. Pöthkow and H.-C. Hege. “Nonparametric models for uncertainty visualization”. In: *Computer Graphics Forum*. Vol. 32. 3pt2. Wiley Online Library. 2013, pp. 131–140 (cit. on p. 13).

[99] K. Potter, P. Rosen, and C. R. Johnson. “From quantification to visualization: A taxonomy of uncertainty visualization approaches”. In: *IFIP Working Conference on Uncertainty Quantification*. Springer. 2011, pp. 226–249 (cit. on p. 8).

[100] S. Rasp and S. Lerch. “Neural Networks for Postprocessing Ensemble Weather Forecasts”. In: *Monthly Weather Review* 146.11 (2018), pp. 3885–3900 (cit. on p. 16).

[101] M. Rautenhaus, M. Böttinger, S. Siemen, R. Hoffman, R. M. Kirby, M. Mirzargar, N. R′ober, and R. Westermann. “Visualization in Meteorology - A Survey of Techniques and Tools for Data Analysis Tasks”. In: *IEEE Transactions on Visualization and Computer Graphics* PP.99 (2017) (cit. on p. 9).

[102] M. Rautenhaus, C. M. Grams, A. Schäfler, and R. Westermann. “Three-dimensional visualization of ensemble weather forecasts - Part 2: Forecasting warm conveyor belt situations for aircraft-based field campaigns”. In: *Geoscientific Model Development* 8.7 (2015), pp. 2355–2377 (cit. on pp. 9, 27).

[103] M. Rautenhaus, M. Kern, A. Schäfler, and R. Westermann. “Three-dimensional visualization of ensemble weather forecasts - Part 1: The visualization tool Met.3D (version 1.0)”. In: *Geoscientific Model Development* 8.7 (2015), pp. 2329–2353 (cit. on pp. 9, 27).

[104] C. Reinbold, A. Kumpf, and R. Westermann. “Visualizing the Stability of 2D Point Sets from Dimensionality Reduction Techniques”. In: *Computer Graphics Forum*. Wiley Online Library. 2019 (cit. on pp. 4, 5, 13, 41).

[105] O. Ronneberger, P. Fischer, and T. Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241 (cit. on pp. 16, 17, 30).

[106] T. Ropinski, S. Oeltze, and B. Preim. “Survey of glyph-based visualization techniques for spatial multivariate medical data”. In: *Computers & Graphics* 35.2 (2011), pp. 392–401 (cit. on p. 8).

[107] H. Saikia and T. Weinkauf. “Global Feature Tracking and Similarity Estimation in Time-Dependent Scalar Fields”. In: *Computer Graphics Forum* 36.3 (2017), pp. 1–11 (cit. on p. 15).

[108] H. Saikia, H.-P. Seidel, and T. Weinkauf. “Fast Similarity Search in Scalar Fields using Merging Histograms”. In: *Topological Methods in Data Analysis and Visualization*. Springer International Publishing, 2015, pp. 121–134 (cit. on p. 15).

[109] M. S. Sajjadi, B. Schölkopf, and M. Hirsch. “Enhancenet: Single image super-resolution through automated texture synthesis”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4491–4500 (cit. on p. 30).

[110] R. Samtaney, D. Silver, N. Zabusky, and J. Cao. “Visualizing features and tracking their evolution”. In: *Computer* 27.7 (July 1994), pp. 20–27 (cit. on p. 15).

[111] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead. “Noodles: A tool for visualization of numerical weather model ensemble uncertainty”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1421–1430 (cit. on p. 10).

[112] N. Sauber, H. Theisel, and H. Seidel. “Multifield-graphs: An approach to visualizing correlations in multifield scalar data”. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), pp. 917–924 (cit. on p. 14).

[113] D. Silver and X. Wang. “Tracking and visualizing turbulent 3D features”. In: *IEEE Transactions on Visualization and Computer Graphics* 3.2 (Apr. 1997), pp. 129–141 (cit. on p. 15).

[114] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations*. 2015 (cit. on p. 30).

[115] B. S. Sohn and C. Bajaj. “Time-varying contour topology”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.1 (Jan. 2006), pp. 14–25 (cit. on p. 15).

[116] C. O. S. Sorzano, J. Vargas, and A. P. Montano. “A survey of dimensionality reduction techniques”. In: *arXiv preprint arXiv:1403.2877* (2014) (cit. on p. 12).

[117] A. Strehl and J. Ghosh. “Cluster ensembles—a knowledge reuse framework for combining multiple partitions”. In: *Journal of machine learning research* 3.Dec (2002), pp. 583–617 (cit. on p. 12).

[118] J. Sukharev, C. Wang, K.-L. Ma, and A. Wittenberg. “Correlation study of time-varying multivariate climate data sets”. In: *Proceedings of IEEE Pacific Graphics 2009*. 2009, pp. 161–168 (cit. on p. 14).

[119] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume”. In: *CVPR*. 2018 (cit. on pp. 16, 17, 31).

[120] J. Thomson, E. Hetzler, A. MacEachren, M. Gahegan, and M. Pavel. “A typology for visualizing uncertainty”. In: *Visualization and Data Analysis 2005*. Vol. 5669. International Society for Optics and Photonics. 2005, pp. 146–157 (cit. on p. 7).

[121] R. D. Torn and G. J. Hakim. “Ensemble-based sensitivity analysis”. In: *Monthly Weather Review* 136.2 (2008), pp. 663–677 (cit. on pp. 2, 24).

[122] J. W. Tukey. *Exploratory data analysis*. Vol. 2. Reading, Mass., 1977 (cit. on p. 8).

[123] J. W. Tukey and P. A. Tukey. “Some graphics for studying four-dimensional data”. In: *The Collected Works of John W. Tukey: Graphics: 1965-1985* 5 (1988), p. 171 (cit. on p. 8).

[124] S. Vega-Pons and J. Ruiz-Shulcloper. “A survey of clustering ensemble algorithms”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 25.03 (2011), pp. 337–372 (cit. on p. 12).

[125] P. Vijaya, M. N. Murty, and D. Subramanian. “Leaders–Subleaders: An efficient hierarchical clustering algorithm for large data sets”. In: *Pattern Recognition Letters* 25.4 (2004), pp. 505–513 (cit. on p. 21).

[126] P. Viswanath and V. S. Babu. “Rough-DBSCAN: A fast hybrid density based clustering method for large data sets”. In: *Pattern Recognition Letters* 30.16 (2009), pp. 1477–1488 (cit. on p. 21).

[127] P. Viswanath and R. Pinkesh. “l-dbscan: A fast hybrid density based clustering method”. In: *18th International Conference on Pattern Recognition (ICPR’06)*. Vol. 1. IEEE. 2006, pp. 912–915 (cit. on p. 21).

[128] K.-C. Wang, K. Lu, T.-H. Wei, N. Shareef, and H.-W. Shen. “Statistical visualization and analysis of large data using a value-based spatial distribution”. In: *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE. 2017, pp. 161–170 (cit. on p. 13).

[129] J. Wang, S. Hazarika, C. Li, and H. Shen. “Visualization and Visual Analysis of Ensemble Data: A Survey”. In: *IEEE Transactions on Visualization and Computer Graphics* (2018), pp. 1–1 (cit. on p. 9).

[130] J. H. Ward Jr. “Hierarchical grouping to optimize an objective function”. In: *Journal of the American statistical association* 58.301 (1963), pp. 236–244 (cit. on p. 11).

[131] M. O. Ward. “A taxonomy of glyph placement strategies for multidimensional data visualization”. In: *Information Visualization* 1.3-4 (2002), pp. 194–210 (cit. on p. 8).

[132] S. Weiss, M. Chu, N. Thuerey, and R. Westermann. “Volumetric Isosurface Rendering with Deep Learning-Based Super-Resolution”. In: *IEEE Transactions on Visualization and Computer Graphics* (2019), pp. 1–1 (cit. on p. 16).

[133] R. T. Whitaker, M. Mirzargar, and R. M. Kirby. “Contour Boxplots: A Method for Characterizing Uncertainty in Feature Sets from Simulation Ensembles”. In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2713–2722 (cit. on p. 10).

[134] W. Widanagamaachchi, C. Christensen, V. Pascucci, and P. T. Bremer. “Interactive exploration of large-scale time-varying data using dynamic tracking graphs”. In: *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*. Oct. 2012, pp. 9–17 (cit. on p. 15).

[135] D. S. Wilks. *Statistical Methods in the Atmospheric Sciences*. 2nd ed. International Geophysics 91. Academic Press, 2005 (cit. on p. 11).

[136] X. Zhu, H. Hu, S. Lin, and J. Dai. “Deformable convnets v2: More deformable, better results”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9308–9316 (cit. on p. 31).

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TVCG.2017.2745178

Visualizing Confidence in Cluster-based Ensemble Weather Forecast Analyses

Alexander Kumpf, Bianca Tost, Marlene Baumgart, Michael Riemer, Rüdiger Westermann, and Marc Rautenhaus

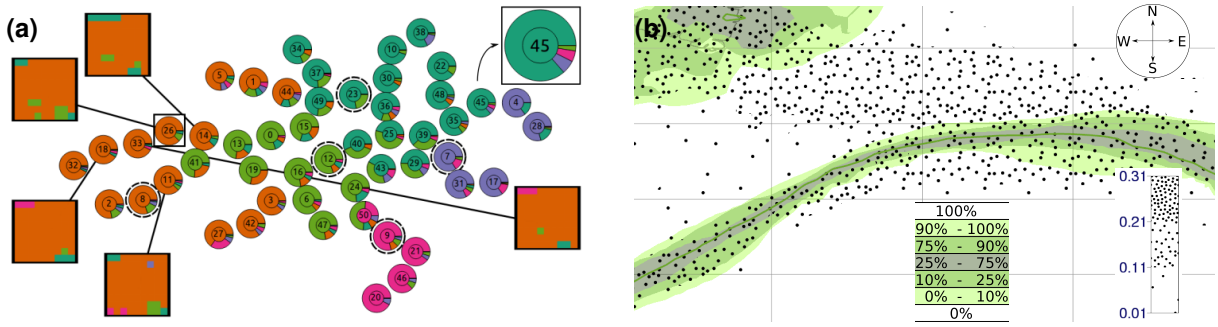

Fig. 1: **(a)** Analysis of the variation in cluster membership over 81 different clusterings of the case "Tropical Cyclone Karl", an ensemble of 51 potential vorticity fields. Circular elements represent ensemble members, colors distinguish clusters (member 45 is enlarged: color of inner circles denotes reference cluster, surrounding pie-charts show how often the member was grouped into another cluster). Dashed outlines highlight cluster representative members. Member 26 is picked, for all members with similar cluster membership variation "variability matrix plots" (squared elements encoding cluster membership of all 81 clusterings) pop up. **(b)** A "contour probability plot" (CPP, different greens show probabilities for contour line occurrence) shows the variability of an iso-contour within a selected cluster. Overlaid stipple pattern shows the spatial variation of the plot with respect to the 81 clusterings.

Abstract— In meteorology, cluster analysis is frequently used to determine representative trends in ensemble weather predictions in a selected spatio-temporal region, e.g., to reduce a set of ensemble members to simplify and improve their analysis. Identified clusters (i.e., groups of similar members), however, can be very sensitive to small changes of the selected region, so that clustering results can be misleading and bias subsequent analyses. In this article, we –a team of visualization scientists and meteorologists– deliver visual analytics solutions to analyze the sensitivity of clustering results with respect to changes of a selected region. We propose an interactive visual interface that enables simultaneous visualization of a) the variation in composition of identified clusters (i.e., their robustness), b) the variability in cluster membership for individual ensemble members, and c) the uncertainty in the spatial locations of identified trends. We demonstrate that our solution shows meteorologists how representative a clustering result is, and with respect to which changes in the selected region it becomes unstable. Furthermore, our solution helps to identify those ensemble members which stably belong to a given cluster and can thus be considered similar. In a real-world application case we show how our approach is used to analyze the clustering behavior of different regions in a forecast of "Tropical Cyclone Karl", guiding the user towards the cluster robustness information required for subsequent ensemble analysis.

Index Terms—Uncertainty visualization, ensemble visualization, clustering, meteorology

1 Introduction

In operational weather forecasting and atmospheric research, cluster analysis of ensemble weather prediction data is frequently used as a tool to analyze forecast uncertainty. Applied to scalar fields of the members of an ensemble forecast, cluster analysis groups together members with similar spatio-temporal development not known in advance [70]. One core objective in such analyses is the determination of representative weather scenarios, i.e., the trends, within a given region of the atmosphere. This region can be selected based on its geographic location (e.g., the operationally computed clustering by the European Centre for Medium-Range Weather Forecasts (ECMWF) uses a European-Atlantic region [13]), or such that it contains a specific weather event of interest, e.g., a trough or a cyclone. Analysis of the computed clusters enables scientists and operational forecasters to distinguish between likely (i.e., forecast by many ensemble members), and unlikely trends. At the same time, by restricting to few representative scenarios, a subsequent analysis can often be sped up significantly.

Before using the cluster information as a basis for further analyses, however, an important question to be resolved is how much confidence can be put into the clustering result. Relying on a determined clustering introduces two major sources of uncertainty: a) The clustering cannot well identify the major trends in the ensemble. A possible cause can be that a region may have been chosen such that the atmospheric features of interest are not entirely covered by this region in all ensemble members, while at the same time features not of interest may be covered that consequently affect the clustering result. Another cause can be that the chosen number of clusters is not sufficient to capture all major trends. This raises the question (Q1) how much "value" is contained in the identified clusters in terms of the similarity of members within a cluster and the distinctness between the different identified clusters. b) The clustering may be sensitive to small changes in the selected region. This requires to answer the question (Q2) how robust (i.e., representative)

- *Alexander Kumpf, Bianca Tost, Rüdiger Westermann, Marc Rautenhaus are with the Computer Graphics & Visualization Group, Technische Universität München, Garching, Germany. E-mail: {alexander.kumpf, bianca.tost, westermann, marc.rautenhaus}@tum.de*
- *Marlene Baumgart and Michael Riemer are with the Institute of Atmospheric Physics, Johannes Gutenberg Universität Mainz, Mainz, Germany. Email: {mbaumga, mriemer}@uni-mainz.de*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

the clustering is with respect to changes of the region, i.e., how strongly do clusters vary when the region is changed. Without resolving these questions, any analysis based upon the obtained clustering cannot be assumed reliable. In the context of ensemble forecasts, however, we are not aware of studies that address these questions in a scientifically rigorous way. Often a merely tacit assumption is made that a "useful" clustering result has been computed and that it is safe to interpret subsequent analyses based on this clustering.

If it is found that a clustering is not robust, the domain expert needs to analyze the variation to understand its cause and to judge its "severeness" to find out if the clustering still contains useful information. By considering cluster membership of single members over multiple clustering results obtained from small changes of the region and identifying members with similar membership variations, sub-clusters with similar behavior as well as robust trends can be identified. The "severeness" of the cluster assignment instability can be judged by investigating its effect on per-cluster quantities (e.g., the cluster mean) that are interpreted in a subsequent analysis. Finally, the domain expert can interpret the obtained confidence information by using additional domain knowledge concerning the physical models and processes represented by the ensemble, and investigate the causes of observed cluster variations, inter-cluster relations, and member similarities and variations.

1.1 Contribution

We introduce novel visual analytics solutions for evaluating the confidence of an obtained clustering result; an overview of the proposed workflow is provided in Sect. 3 and Fig. 2. An artificial data set with known characteristics is used to motivate and introduce the visual encodings and interaction techniques we propose with regard to the requirements of the application task (Sect. 4). We apply our solutions to analyze a real-world ensemble forecast that predicted "Tropical Cyclone Karl" in 2016, and demonstrate that the visualization content we provide generates new and important insights (Sect. 5).

To resolve Q1, we provide a combination of abstract cluster-centric views with a linked map (Fig. 2c). The displays enable comparative visualization of the similarity of ensemble members within an identified cluster and the similarity of the members in the unclustered ensemble. In particular, we display where geographically the standard deviation (STDEV) of the clustered ensemble is reduced compared to the STDEV of the unclustered ensemble.

To resolve Q2 and the follow-up robustness analysis, we provide a set of interactive and linked cluster-centric, member-centric, and spatial views (Fig. 2d). A "cluster-centric robustness display" (CRD) visualizes changes of the clusters relative to a reference clustering (e.g., the clustering result for the initially chosen region). This view shows at a glance whether a clustering is stable or whether it changes significantly when the region is changed. Furthermore, the CRD communicates how many members change their cluster membership, and into which clusters these members change. The CRD is linked to a "member-centric robustness display" (MRD). It uses circular pie-chart diagrams for each ensemble member, arranged in a 2D coordinate system spanned by the first two major principal components (PCs) of the ensemble (Fig. 1a). We focus in particular on the requirement to visualize the cluster membership variation for each ensemble member when the clustering region is changed, so that the frequency of membership changes per member can quickly be perceived. The user can pick a member in the MRD to let the interface generate a matrix plot for this member in which the changes in cluster membership for each chosen region are shown. Simultaneously, matrix plots of all members having a similar cluster membership variation are displayed. This provides a very intuitive mechanism to instantly find sub-clusters of ensemble members that behave in a similar way.

Finally, we propose a cluster-centric view combining a variation of contour boxplots [68] and contour variability plots [15] with an overlaid stipple pattern. These "contour probability plots" (CPPs) (Fig. 1b) indicate the point-wise probability that within a selected cluster an isovalue is exceeded; stippling is used as a distinct visual channel to emphasize variations due to changes in the clustered region.

2 Related Work

Clustering analysis is a well established statistical tool to identify groups of samples in a dataset that are similar with respect to a similarity measure. General introductions to data clustering are provided in the books by, e.g., Everitt et al. [12] and Gan and Ma [21]. For the presented work, related topics of particular relevance are the application of cluster-based analysis in meteorology, cluster-based ensemble visualization, and visual methods to judge the robustness of clustering results.

2.1 Cluster-based analysis in meteorology

A general overview of common clustering techniques and their application was provided by Wilks [70], a number of cluster visualization techniques for clustered climate data (no ensembles) were discussed by Nocke et al. [45]. In weather forecasting, clustering is commonly used in a static, pre-defined way. A representative example is the operational clustering of ensemble members at ECMWF [13], where forecast scalar fields of geopotential height are clustered in three different time windows for a static data region covering the ECMWF member states. Results are visualized in static matrix plots containing small forecast maps of the cluster representatives [13]. In meteorological research, ensemble cluster analysis has been used to improve understanding of various aspects of atmospheric predictability. For example, weather-regime related predictability was studied by using k-means clustering of jet wind profiles and comparing identified forecast regime transitions to observed climatological probability [17]. Clustering of leading PCs was used to study forecast scenarios in relation to tropical cyclones [1, 22, 30] and a heat wave that occurred in 2010 [53].

Clustering results were sometimes displayed by color-coded points in 2D PC plots [22, 53]. Different forecast scenarios were illustrated by cluster mean [1, 22] or by representative cluster members [30, 53]. Harr et al. [22] provided subjective criteria for the optimal number of clusters based on a discussion of the underlying meteorological charts. The sensitivity of the analysis to the number of PCs used in clustering is briefly discussed based on (subjective) changes in the cluster mean. Sensitivities with respect to the choice of the analysis domain are not discussed. In case of bifurcation-type behavior, clusters are often defined ad-hoc, for example as "good" and "bad" forecasts (e.g., [41,66]). Statistically significant differences in the cluster mean of meteorological variables are sought to explain the forecast bifurcation.

2.2 Cluster-based ensemble visualization

With respect to visualization research, our approach is related to techniques for ensemble visualization – a sub-field of uncertainty visualization, for which a number of surveys exist [3, 40, 48, 52]. Uncertainty in scientific data is often estimated by means of ensembles – a set of representative realizations of a simulated phenomenon, obtained from simulations with different initial conditions and/or physical models. Such data is typically spatiotemporal, multivariate, and multivalued [29, 37], making the analysis and visualization processes difficult. Several methods have been proposed to reduce this complexity, e.g., by visualizing statistical summaries including mean and STDEV with color maps, contours, surface deformation, opacity, and variations of boxplots [11, 25, 37, 49, 51].

Clustering is another approach to reduce the complexity of ensemble data. Bordoloi et al. [4] proposed realization- and distribution-based hierarchical clustering to reduce the amount of information to be visualized. Bruckner and Möller [6] used density-based clustering to identify similar volumetric time sequences in physically-based ensemble simulations. Beham et al. [2] used hierarchical clustering to group similar geometric shapes. Reh et al. [58] clustered similar pores in industrial XCT data into mean objects (Mobjects) and then visualized the per-voxel probability of belonging to an Mobject using transfer functions. Hummel et al. [26] clustered using Minimum Spanning Trees for trend analysis, to compare the material transport in flow ensembles. Other techniques have used clustering to group iso-contours in scalar fields [8, 64] or streamlines in flow fields [47]. Bruckner and Möller [5] proposed the use of signed distance functions to analyze different iso-contours of the same scalar field, Rathi et al. [55] used such functions for shape analysis, and Ferstl et al. [14, 15] demonstrated

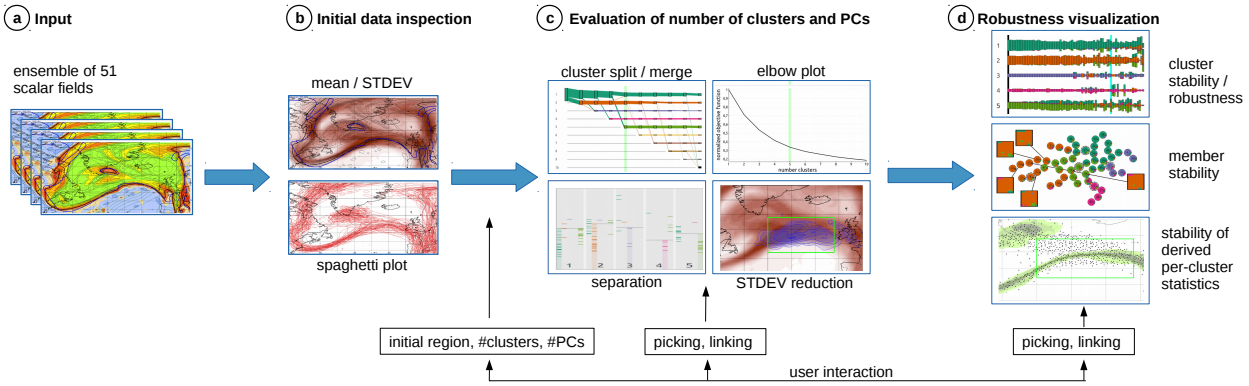

Fig. 2: Method overview. **(a)** Input is an ensemble of 2D scalar fields, which is **(b)** first inspected using statistical summaries and spaghetti plots. The user selects the region to be clustered and **(c)** uses different visualizations to interactively select a suitable number of clusters and of principal components to be used for clustering. Finally, **(d)** by means of linked views in which specific clusters and ensemble members can be selected, the value in a specific clustering and the robustness of clusters with respect to changes in the input region are analyzed.

the use of clustering to find visual abstractions for conveying the major trends in ensembles of trajectories and iso-contours. Recently, Ferstl et al. [16] proposed the use of clustering to detect similarities in the temporal variation of ensemble members.

2.3 Quality and robustness of clustering results

Common techniques to judge the quality and robustness of clustering results are surveyed in the book by Everitt et al. [12, Ch. 9]. Graphical depictions include banner plots, silhouette plots, and stripes plots (the latter are also used in our workflow). García-Escudero et al. [20] review robust clustering methods.

For the visualization of single clusterings, a number of standard techniques are known, including dendrograms to show hierarchical clusterings, heatmaps and clustergrams [61, 69], 2D principal component plots [44], and graph-based visualization techniques. Most commonly, clusters are visualized by giving each data point a location in a 2D map, so that similar data points are grouped together. For high-dimensional data, dimensional reduction is typically used to find appropriate locations [7, 24, 39]. Another option is to display clusters by means of parallel coordinates [23, 27]; a number of techniques have been proposed to extend parallel coordinates towards the visualization of clusters [19, 38, 50, 72], including the combination with heat maps [46] and radial tree displays [2, 32]. Lex et al. [36] introduce the Caleydo Matchmaker technique to visualize clusterings of subgroups of data using special parallel coordinate plots. This framework can also be used to compare different clustering strategies.

Only few approaches, to the best of our knowledge, combine the output of several clusterings into one; examples include the "cluster ensembles" by Strehl and Ghosh [62]. They applied different methods to obtain different clustering outcomes, and generated a single clustering from all outcomes. The probability accumulation by Wang et al. [67] aggregated multiple clustering outputs into a new one.

3 Workflow Overview

Our proposed visual analytics solution enables meteorologists to interactively analyze the confidence in clustering results obtained from a selected spatial region. Fig. 2 shows an overview of the proposed workflow. Input (Fig. 2a) is a set of scalar forecast fields representing physical quantities including air temperature and potential vorticity (PV, a quantity computed from vorticity and stratification of an air mass; frequently employed to analyze the dynamics of weather systems). In this work, we consider scalar field data from the ECMWF Ensemble Prediction System (ENS; e.g., [35]). The ensemble comprises 50 perturbed members and an unperturbed control forecast (that is started from the "best" initial conditions). Past forecast data is available from the TIGGE archive [63] on a regular longitude–latitude grid in the horizontal; in the vertical, data is available on levels of constant pressure.

The user first inspects the data by means of non-cluster-based visualizations of the ensemble data (Fig. 2b). We provide maps of ensemble mean and STDEV, and spaghetti plots of relevant features including isocontours. This initial inspection is used to select the data region over which the clustering should be performed. Additionally, the user defines a range of numbers of clusters that are subsequently evaluated.

Following common practice in meteorology (cf. the operational ECMWF clustering [13] and research studies [1, 22]), our meteorological collaborators have explicitly requested the application of principal component analysis (PCA; e.g., [28]) to reduce input data dimensionality and k-means clustering. For the user-selected parameters, different types of diagrams facilitate determination of the optimal number of clusters suitable for separating major trends in the data (Fig. 2c). Cluster split-merge diagrams (CSMD) and "elbow plots" [65] indicate the number of clusters beyond which no significant gain in the cluster-specific objective function is achieved. "Stripes plots" [34] and displays of the reduction of STDEV achieved by a clustering convey how well clusters are separated and how well similar members have been identified (to resolve Q1).

For the determined number of clusters and the corresponding clustering (the "reference clustering"), a further CSMD visualizes changes in the clusters relative to the reference clustering when the number of used principal components (PCs) is changed. By default, as many PCs are used as to explain 80% of the variance of the data (default at ECMWF [13]; yielding on the order of 5 to 15 PCs), yet to judge robustness with respect to the number of PCs, the user is concerned with finding the smallest number of PCs that explain sufficient variance such that the clustering result does not change if more PCs are added.

Finally, with the selected region, number of clusters, and number of PCs, linked cluster- and member-centric views facilitate investigation of Q2, the robustness of the clustering with respect to changes in the selected region (Fig. 2d). If the clustering is found to be not robust, the user can focus on a particular cluster or ensemble member and visually analyze variation in cluster composition and membership to investigate the questions of: (Q3a) Which clusters are robust and which are not robust, and under which region changes are ensemble members changing cluster assignment? (Q3b) Which ensemble members are robust and which are changing ensemble assignment, and are subgroups of ensemble members changing in a similar way? (Q4) What are the effects of these changes on per-cluster quantities that are interpreted in a subsequent analysis?

4 Visualization Techniques

We motivate and introduce details of our interactive visual analysis workflow using a synthetic dataset that illustrates data characteristics that our users need to be able to analyze. This dataset with known clustering allows, e.g., to intentionally perform robust and non-robust

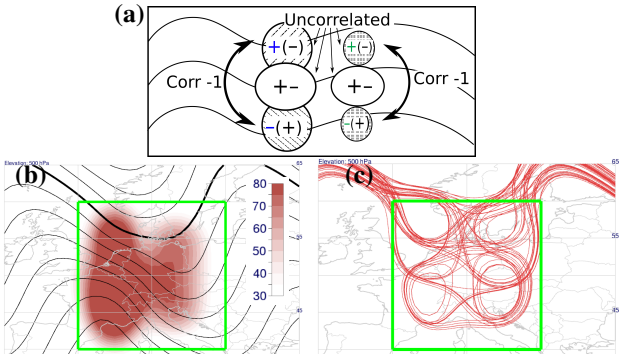

Fig. 3: The synthetic dataset. **(a)** Each member contains two dipoles (hedged) and in between each one a monopole. The polarities for all 4 poles are randomly chosen for each member. Neither **(b)** ensemble mean (contour lines) and STDEV of geopotential height in meters (color) nor **(c)** spaghetti plot of the thick black contour line in (b) indicate the dataset's features or a suitable number of clusters. The green box is chosen as clustering region.

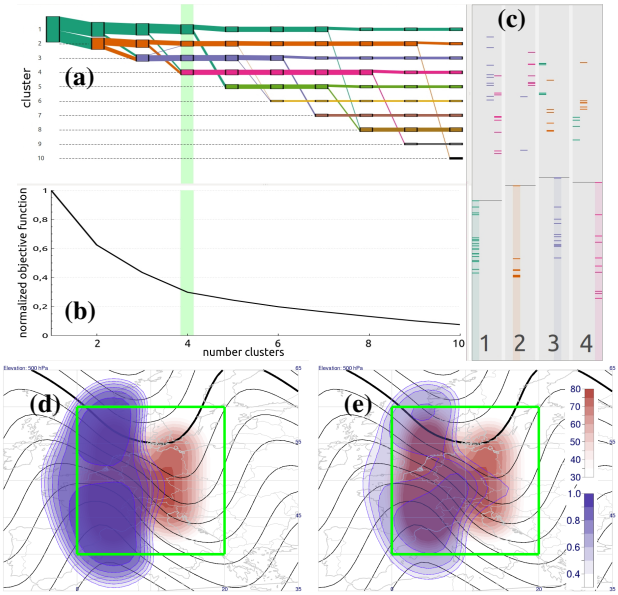

Fig. 4: Visualizations to judge clustering value for the synthetic case. **(a)** Cluster split-merge diagram (CSMD) for increasing number of clusters. **(b)** Elbow plot. **(c)** Stripes plot showing separation of clusters for the clustering selected in (a) (highlighted by green bar). **(d)** Reduction (blue) in fraction of STDEV (red, in meters) for four clusters is significantly improved compared to **(e)** three clusters.

parameter changes in order to demonstrate correct identification by our methods.

4.1 Synthetic dataset as guiding example

The synthetic 2D dataset was generated with the same data modality as the real-world ECMWF ENS forecasts, comprising 51 members. We use a horizontal grid-spacing of 1° in longitude and latitude. Fig. 3a illustrates the patterns contained in the dataset. The individual ensemble members contain a wave-like pattern (the same in all members) that mimics the wave-patterns typically encountered in atmospheric data. In each member, noise is added to slightly distort the waves.

Clustering information is added by constructing features that uniquely group the members but are not discernible in plots of ensemble mean, STDEV or in spaghetti plots. Four independent features –two dipoles and two monopoles– are added to each member with a randomized polarity for each feature and member. The dipole located in the western part adds a stronger disturbance to the background wave than the eastern dipole. It is hence expected to dominate the clustering result when covered by the clustering domain. The uncorrelated monopoles are centered between the northern and southern parts of each dipole to further disturb the field so that an increased number of clusters can be identified.

In total, 16 different cases are generated, randomly distributed over the 51 members. However, due to the dominance of the western dipole and the high degree of non-correlation between the features, fewer clusters are expected to be detected by clustering. For instance, if both dipoles are covered by the cluster region, it is unlikely that a clustering result can separate both signals.

Figs. 3b, c show ensemble mean and STDEV, as well as a spaghetti plot for the initial data inspection. In the ensemble mean field, the features average out; the ensemble STDEV does not separate distinct features (Fig. 3b). The spaghetti plot of a contour line shows high uncertainty but gives no indication for the existence or number of clusters (Fig. 3c). Based on these plots, a user may choose the cluster region shown in Fig. 3b, covering the region containing high uncertainty.

4.2 Visualizing clustering "value"

Fig. 4 shows the interactive linked views we provide to determine the most suitable number of clusters by judging the value of the computed clusterings for different numbers (Q1). Here it is important that quantitative (about the number of members per cluster) and qualitative (about the spread of the members per cluster) information is provided, so that the domain expert can analyze which ensemble members belong robustly to a certain trend, and how representative the trend indicated by a certain cluster is. This information allows judging which clusters can be relied upon in the following meteorological application (Q4).

An abstract CSMD (Fig. 4a) shows how clusters split as the number of clusters is increased and depicts where members change into different clusters. Each cluster is visualized by a vertical bar with constant width and unique color. A bar's height represents the number of ensemble members contained in the corresponding cluster. Connecting bands between bars in subsequent clusterings indicate the number of ensemble members that stay in the same cluster or leave a cluster and merge into a different one. The view is augmented by an elbow plot (cf. Thorndike [65]), a simple plot of how the objective function minimized by the clustering algorithm decreases as the number of clusters is increased (Fig. 4b). The elbow plot indicates a suitable number of clusters; the point of the strongest "bend" in the curve can be interpreted as a trade-off between a small number of clusters and a small objective function. Unfortunately, the shape of the curves do not always allow this bend to be unambiguously identified.

For the synthetic use case (Fig. 4), we compute clusterings using $k = 2, \ldots, 10$ clusters. Evaluation of the elbow plot indicates 4 clusters. The CSMD shows that for 5 clusters only the first cluster is split and all other clusters stay the same. For even more than 5 clusters we judge the further decrease of the objective function to be insufficient. So, we choose 4 clusters and keep in mind that the first cluster could be split into two if it does not show a clear trend.

To further evaluate a specific clustering, the user can select this clustering, i.e., by moving the green vertical line in the split-merge-diagram to this clustering. Upon movement, a linked stripes plot (Fig. 4c) is updated and displays specific information for this clustering. The stripes plot provides an abstract view on cluster separation; it has been proposed by Leisch [34] and shows the distance of each member to its cluster center and the distance to its second closest cluster center. If those two distances differ only little, the clusters are poorly separated, indicating that a different number of clusters may yield better results. The stripes plot, which notably does not provide any robustness information, indicates good separation for all numbers of clusters in the synthetic case, so in this case, it does not argue against any. For real world cases, clusters are usually less separated (e.g., clusters 1 and 5, Fig. 13). Linked spatial map views (Fig. 4d, e) provide further information on spatial characteristics of a selected clustering,

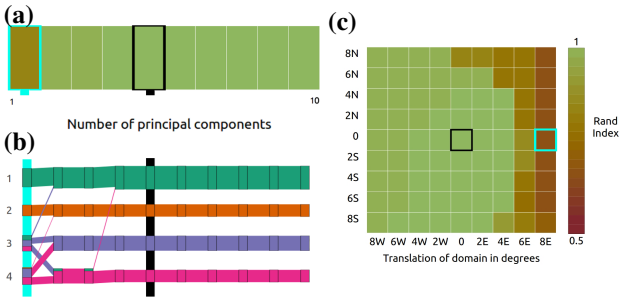

Fig. 5: Visualizations of cluster robustness. **(a)** Rand index and **(b)** CSMD visualizing the robustness with respect to changes in the number of principal components (PCs). For the synthetic case, using more than 4 PCs does not lead to further changes. **(c)** Rand matrix shows high sensitivity (red) to translations of the region to the east.

i.e., about the spread of the members in one cluster. A central quantity in our workflow is the reduction of STDEV per model grid-point [60], providing direct information about the average compactness of a cluster (Fig. 4d, e). STDEV reduction represents the ratio of STDEV within each cluster to the STDEV of the entire ensemble, thereby summarizing how much the identified clustering has grouped similar members into clusters. It can be computed for individual clusters and for the set of all clusters; in our workflow we display the total reduction of all clusters. STDEV reduction is computed as

$$r_k = 1 - \sqrt{\frac{f_k(\mathscr{S})}{f_1(\mathscr{S})}} \tag{1}$$

where f_k is the same as the k-means objective function

$$f_k(\mathscr{S}) = \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2, \quad \{S_1, \ldots, S_k\} \in \mathscr{S}, \tag{2}$$

which is the sum of the intra-cluster sums of the squared errors. S_i denotes the set of all members in cluster i, x a member in cluster S_i, μ_i the mean of cluster S_i. $\mathscr{S}$ is the set of all disjoint partitions of members $\{1, \ldots, n\}$ into k clusters. The k-means algorithm minimizes Eq. 2 globally, it hence makes sense to use this measure for per-grid-point calculations as well. However, other measures exist which can be used for different clustering algorithms [31].

For the synthetic use case, STDEV reduction confirms four clusters to be appropriate (Fig. 4d); it is significantly improved compared to three clusters (Fig. 4e) but only marginally worse compared to five clusters (not shown).

4.3 Visualizing clustering robustness

Figs. 5 and 6 show the visualizations we provide to judge the robustness of the obtained clustering (Q2). In a first step, for the determined number of clusters and the corresponding clustering (the reference clustering), a plot of the Rand index [54] and a CSMD (Fig. 5a, b) are used to visualize changes in the clusters relative to the reference clustering when the number of used PCs is changed. The Rand index is a central summary measure to compare clusterings with differing input parameters; it measures agreement and disagreement of all pairs of members in two partitions. An index of 1 implies identical, an index of 0 maximally distinct clusterings. An index of 0.5 already indicates large changes; we clamp the transfer function at this value. Wu et al. [71] compared external clustering validity measures for k-means and recommended the use of the Rand index among others. For the synthetic case, 4 PCs are sufficient since more do not lead to changes in the cluster assignment anymore.

Next, the user can analyze the robustness of the identified clusters (Q3a, cluster-centric visualization), the robustness of the per-member assignment to specific clusters (Q3b, member-centric visualization), and the robustness of statistical quantities (including spatial fields, Q4)

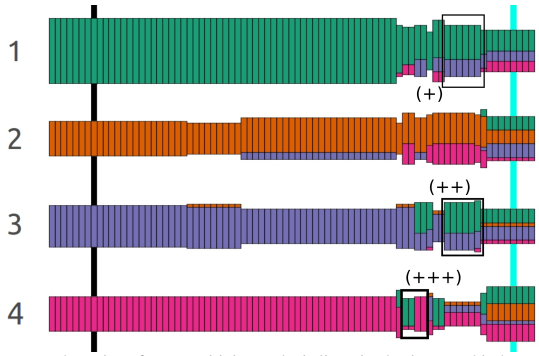

Fig. 6: Cluster-centric robustness display (CRD). The size of the bars represents cluster size, colors represent membership in reference clustering (black background bar). In clusterings (+,++), the blue and green clusters exchange members; in clusterings (+++), cluster 4 changes almost completely. Turquoise background bar corresponds to turquoise clustering in Fig. 5c.

derived per-cluster to summarize cluster statistics. The latter includes robustness of cluster mean and STDEV, and the robustness of trend plots including CPPs.

For the synthetic case we evaluate the robustness of the clustering result with respect to domain translations up to 8° in each cardinal direction, using a 9×9 grid of 2° translations. Fig. 5c shows an example of the Rand index color-mapped to a 2D matrix representing these horizontal and vertical translations. The matrix highlights those region translations which strongly affect the clustering result (Q3a). The Rand matrix shows that for small perturbations the clustering is very robust; in fact, it hardly changes at all. For larger translations towards the east, however, the clustering changes significantly. This change could be caused by outliers, or it can indicate a new feature not captured in the initially selected region that starts to dominate. In the following, we introduce an additional type of diagram to further analyze the changes in clustering results due to region translations.

4.3.1 Cluster-centric robustness display

To provide to the user an overview of cluster changes for different region translations relative to the reference clustering (Q3a), we provide a cluster-centric robustness display (CRD; Fig. 6). The CRD is linked to the Rand matrix, to let the user select a clustering by picking on the corresponding matrix entry, and highlighting the corresponding information in the CRD. The CRD depicts on the horizontal axis the different clusterings for all possible region translations. Cluster sizes are indicated by the height of the bars, the colors show which proportion of which cluster of the reference clustering is contained in the selected clustering. To reduce overlap, the clusters are centered and the total space used by the bars in vertical direction can be adjusted.

To fit two-dimensional parameter changes into this view, a simple linearization of the clustering, or alternatively, an ordering minimizing the number of cluster membership changes, can be selected. To compute this ordering, all permutations of all clusterings are tested, which works well for up to 10 clusters. For more clusters, the Kuhn-Munkres algorithm [33] can be used, for which efficient implementations exist with a runtime complexity of $\mathcal{O}(n^3)$. When reducing the horizontal size of the bars, connections between them become visible which show where and how many members changed from one cluster to another (e.g., in Fig. 5b). Further, any realization can be selected in the CRD and all connected views automatically update using the selected realization. In Fig. 6, it can be seen that many translations do not affect the clustering at all, followed by some only affecting the orange and blue clusters. For extreme translations to the east, completely new cluster compositions arise, indicating that a new feature is starting to dominate.

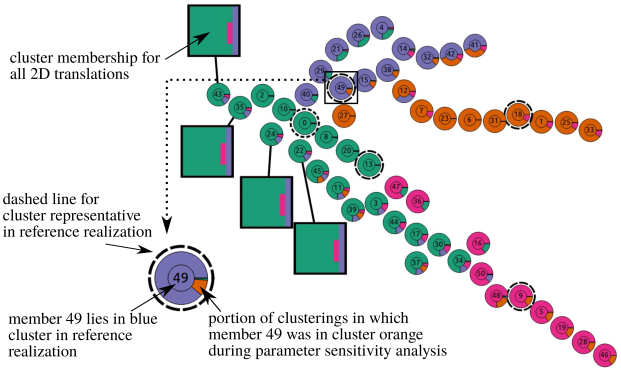

Fig. 7: The member-centric robustness display (MRD) shows a pie-chart for each ensemble member, each encoding robustness information. Stable and unstable members can be identified by means of cluster membership information encoded in the pie-chart. Matrix pop-ups show cluster membership changes with respect to region translations, arranged as in Fig. 5c.

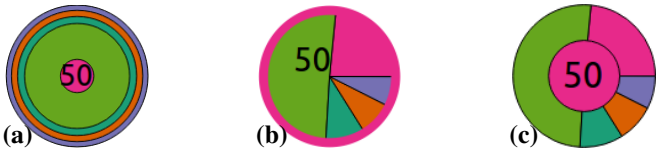

Fig. 8: Possible glyph designs for the MRD. **(a)** Spherical arrangement of membership information. **(b)** Membership information represented by a pie-chart, with reference cluster information surrounding the chart. **(c)** As (b) but reference cluster information in the chart center.

4.3.2 Member-centric robustness display

A further important task is the analysis of the changes in cluster membership of the ensemble members (Q3b). Such an analysis, in particular, has the purpose to identify members that remain together in the same cluster when region translations are performed or the number of clusters or PCs are changed. To visualize this member-centric information, we propose a member-centric robustness display (MRD) using a separate glyph for every member (see Fig. 7 for the synthetic use case).

In the design of the MRD the following criteria were considered: a) Each glyph needs to show simultaneously different types of information, i.e., the cluster of the member in the reference clustering, the member's unique ID, the frequency of changes in cluster membership for the performed region translations, and a detailed view of the relation between cluster membership and region translations. b) A large number of glyphs, i.e., one for each ensemble member, needs to be shown at once to enable a comparative study of members. Due to b), our first design decision was to separate the visual encoding of the latter type of information in a) from the others, to avoid visual overloading and clutter. Our second design decision was with respect to the glyphs shape and structure. In Fig. 8, we show possible circular (to minimize coverage) glyph designs, of which the third one is the design we propose.

In our design, each member is represented by an inner circle and a surrounding pie-chart. The color of the inner circle indicates the reference cluster of the member, and the member ID is shown in the inner circle. The pie-chart shows how often member i was in cluster j, with the frequency sorted decreasingly –but always starting with the reference cluster– and displayed counter-clockwise. If the member is the cluster representative in the reference clustering, a dashed border around the outer circle is added. When looking at the two alternative designs, one can clearly observe the following: In design 8a, where the pie chart is replaced by circular rings depicting the membership variations, the inner circle and rings might become so small and thin that the important information cannot be perceived clearly any more. In design 8b, the member ID obscures some of the pie-slices, and the outer ring encoding the reference cluster can become too thin to be clearly observable (under the constraint that the glyph cannot be arbitrarily large and the pie-cart needs sufficient space).

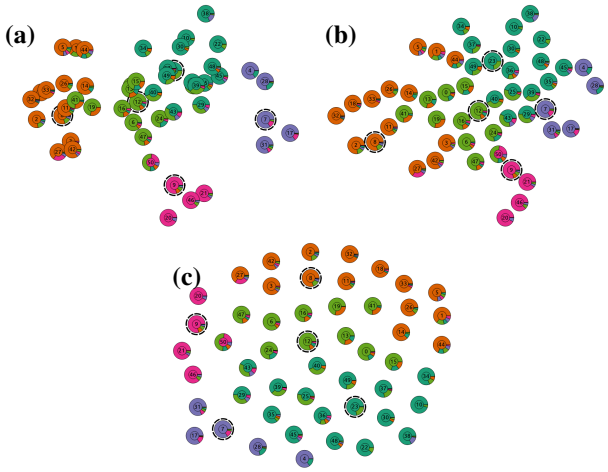

Fig. 9: MRD layouts based on **(a)** the first two PCs, **(b)** as (a) but squeezed and without overlap, **(c)** a force directed graph layout.

A click on a pie-chart opens a matrix plot giving a detailed view of the member-specific relation between cluster membership and region translations. Color in the displayed matrix indicates the member's cluster assignment in all clusterings. Double-clicking on a pie-chart opens matrix plots for all members with similar cluster change characteristics, which can immediately reveal similar patterns across multiple members. The pop-ups are placed around the convex hull of the already drawn pie charts to avoid occlusions. In case the closest pop-up position leads to overlap with another pop-up, it is translated to the next closest position without producing overlap.

For the placement of pie-charts on the 2D canvas, we use the members's locations in the 2D coordinate system spanned by the 1st and 2nd PCs (Fig. 9a). Despite more PCs being used for the clustering in most cases, the first two components provide the most separation between members and, thus, yield least overlap when glyphs are drawn. Nevertheless, overlaps can occur, requiring strategies to avoid them.

A force-directed graph layout as proposed by Fruchtermann and Reingold [18] can be used to enforce that members of the same clusters are placed close to each other. Here, the members are initially placed on a circle, then "pulled" together by using their clustering distance measure as a force. In the resulting plot, the members are placed very homogeneously in the 2D image (Fig. 9c). On the downside, the visual separation of the clusters which is present in Fig. 9a gets lost. Due to this, we perform a different placement strategy, which tries to resolve the overlaps that are introduced by the initial approach. First, the distance between the two pairwise closest members is determined, and these two members are then considered being processed. All remaining members are successively moved in the direction of the processed members until their distance to them is approximately the same as the minimum distance. After all glyphs are shrunk together, the domain is rescaled to fit the available space. The result is shown in Fig. 9b. In the MRD in Fig. 7, there are multiple entirely stable members in the blue and green cluster (e.g., member 0), indicating stable sub-clusters. When picking member 35, the pop-ups display 4 members also changing only for extreme eastward translations. Those 4 members are always clustered together forming a stable sub-scenario.

4.3.3 Robustness of per-cluster summary statistics

Finally, we provide information on the robustness of per-cluster summary statistics, including plots of major-trends and cluster mean and STDEV. To visualize major trends, variability plots [15] and contour boxplots [68] are well suited, however, since they do not show point-

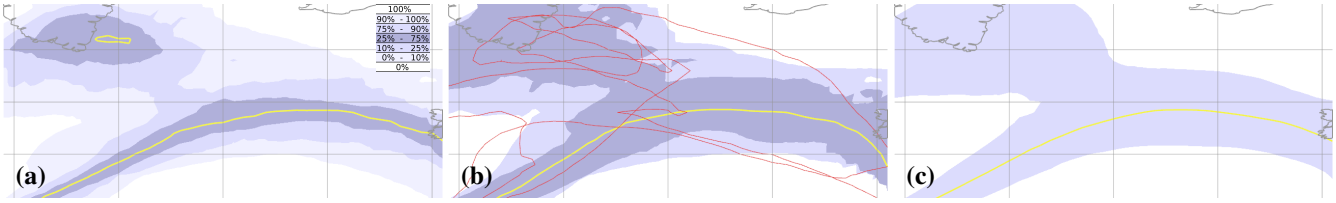

Fig. 10: We propose **(a)** "contour probability plots" (CPPs), lobes enclosed by iso-contours of constant probability of a set of members exceeding the threshold value for which the lobe is shown. The lobes have similar shape as **(b)** contour boxplots [68] (50% inner lobe) and **(c)** variability plots [15] (1.5 STDEV) but are more straightforward to interpret when overlaid with robustness stippling.

wise quantitative information that can be overlaid by clustering robustness information, we propose a variation, which we call "contour probability plots" (CPP). A CPP depicts lobes which indicate the probability that a contour line is locally contained in the lobe. CPPs are generated by grid-point-wise computation of the probability that the considered isocontour threshold is exceeded in the ensemble. Isocontours of this probability field are then used to construct the lobes, e.g., the 25% and 75% isocontours enclose the 50% band. The example in Fig. 10a displays lobes indicating bands for the inner 50%, and outer 15% and 10% on each side. When CPPs for multiple clusters are overlaid, a reduced set of lobes is beneficial to avoid cluttering. A distinct advantage of CPPs is that the sensitivity, calculated as the STDEV of this probability over different clusterings, can be interpreted as the average translation of the contour plot borders. This allows the user to see right away how the contours of a cluster might change when using a different clustering region.

To visualize the robustness of a cluster, i.e., its variation in terms of membership changes, a distinct visual channel that can be combined with CPPs is required. Retchless and Brewer [59] investigated different types of overlays to depict uncertainty in climate variables. Following their study, we use stippling since it can be clearly distinguished from the colored underlay and does not interfere with other communication channels. Our rendering technique follows the texture-based approach presented by Maskey and Newman [42].

Fig. 1b shows the sensitivity of a CPP, Fig. 11 that of STDEV reduction plots via stippling. In Fig. 1b, the stippling shows a large area where the STDEV of the probability is on the order of 0.1–0.2, indicating, e.g., that the lobes could well extend into the white region if the selected region is changed. The displayed cluster characteristics are hence very uncertain. In Fig. 11, a high uncertainty is visible throughout the initially clustered area, indicating that for translated domains the geographical areas in which variance is reduced by clustering also change. For instance, when considering a clustering region translated to the east (Fig. 11b), we see a very different area being clustered "well". Here, the two eastward dipole features were detected. The contour probability plots clearly confirm this detection of the distinct features by different cluster regions (Fig. 11c and d).

5 Results

All visualization techniques, as well as PCA and clustering methods, have been implemented in the open-source meteorological ensemble visualization tool "Met.3D" [43, 57]. Met.3D provides a suitable infrastructure that already included an ensemble data processing pipeline and visualization functionality for meteorological maps. Stippling functionality, as well as all abstract and linked charts presented in this article, have been added in the context of this work. Clustering is implemented using the library "Cluster 3.0" [9].

To compensate for the denser geometric grid-point spacing towards the poles in the regular longitude–latitude grid, each grid-point needs to be weighted by the size of its grid-cell. We follow Ferranti and Corti [13] and weigh each grid point with the cosine of its latitude prior to applying PCA to the data. Clustering and cluster matching for different realizations require most computing resources; on an Intel® Xeon CPU E5-1650 v2 with 6 cores @ 3.50GHz, each cluster run took from a few seconds up to two minutes depending on the data. Cluster runs are executed in parallel, for the presented case study clustering and cluster matching required on the order of 20 minutes. Once the data are clustered, visualization performance is interactive.

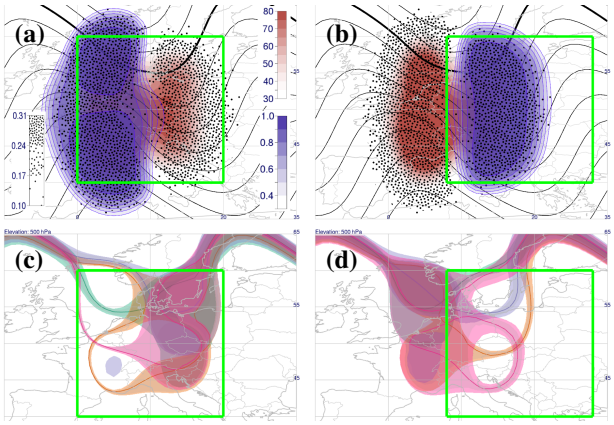

Fig. 11: **(a)** Reduction (blue) in fraction of ensemble STDEV (red, in meters) for the synthetic case, overlaid with robustness due to region translations (stippling shows STDEV of reduction). The intense stippling indicates that region translations can cause clusterings to reduce ensemble STDEV in different regions. **(b)** STDEV reduction caused by a region shifted eastward. **(c, d)** CPPs 85% bands of the respective regions indicate that different features dominated the clusterings.

5.1 Tropical Cyclone Karl

To demonstrate the application and value of our method, we discuss a real-world forecast issued during the North Atlantic Waveguide and Downstream Impact Experiment (NAWDEX [10]), a field campaign in which three of the authors have been involved. Tropical Cyclone Karl crossed the North Atlantic in late September 2016 and was associated with heavy precipitation in Norway. A number of days prior to measurement flights it posed significant difficulties in forecasting due to high uncertainty in the forecasts. A major objective of the data analysis activities of the campaign is an investigation of ensemble behavior and determination of the physical processes that caused the uncertainty.

We consider the ECMWF ENS forecast from 00:00 UTC 22 September 2016 (data are at 1° horizontal resolution, on pressure levels in the vertical and at 6-hourly time steps) and focus on the development of Karl in subsequent days. In this ensemble, very different developments for Karl were predicted. Of critical interest for the analysis of the situation is the interplay between upper-level PV (of particular interest is the 2-PV-units (PVU) isosurface used to represent the tropopause, the boundary between the troposphere and stratosphere), the upper-level jet stream (strong winds in the vicinity of the tropopause), lower-level temperature and moisture fields, and Warm Conveyor Belts (airstreams that lift warm and moist air from near the surface to the upper troposphere; c.f. [56]). In the analysis presented here, we –the meteorological domain experts in the author team– are interested in whether we can determine distinct scenarios (i.e., clusters) in the upper-level PV field to investigate the relationship between PV and lower-level developments, e.g., the low-level potential temperature field.

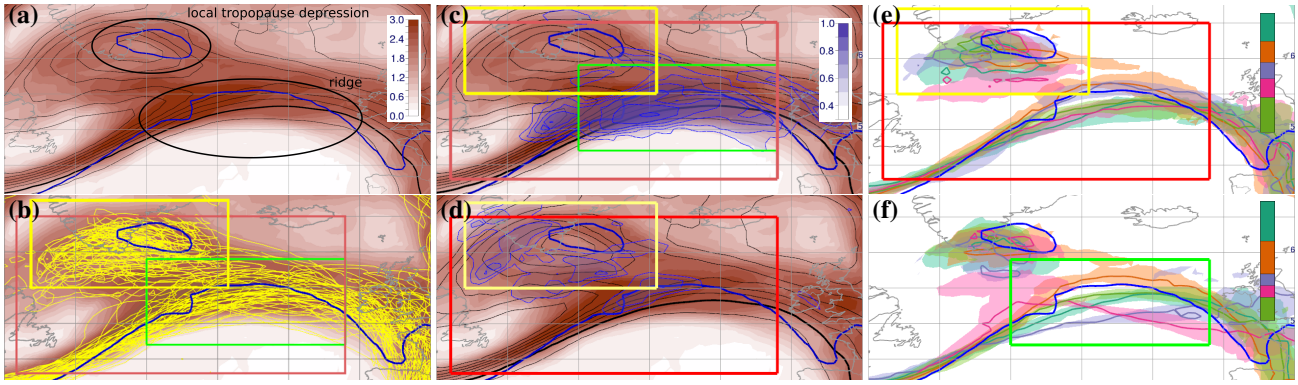

Fig. 12: Case study. A clustering region is chosen based on **(a)** ensemble mean (black contours) and standard deviation (color in PVU) of potential vorticity at 250 hPa, and **(b)** a spaghetti plot of the 2-PVU isocontour. Shown is the ECMWF ENS forecast from 00:00 UTC 22 September 2016, valid at 00:00 UTC 26 September 2016. Thick blue contour shows ground-truth observation; red, green and yellow boxes represent clustering regions. **(c)** STDEV reduction (blue) when clustering the red region. There is almost no STDEV reduction in the yellow region. **(d)** Clustering the yellow region leads to only little STDEV reduction. **(e)** CPP (85% bands) of the 2-PVU isocontour showing clusters of the yellow clustering region. No particularly different scenarios have been identified. **(f)** When clustering the green region, distinct scenarios for the ridge are identified.

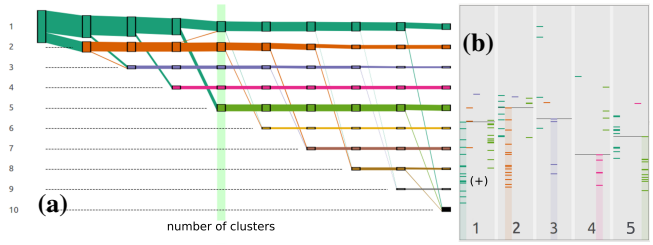

Fig. 13: **(a)** CSMD and **(b)** stripes plot (for five clusters) showing weak separation of clusters 1 and 5 **(+)**. The CSMD shows the creation of a robust cluster 5 when increasing the number of clusters from four to five.

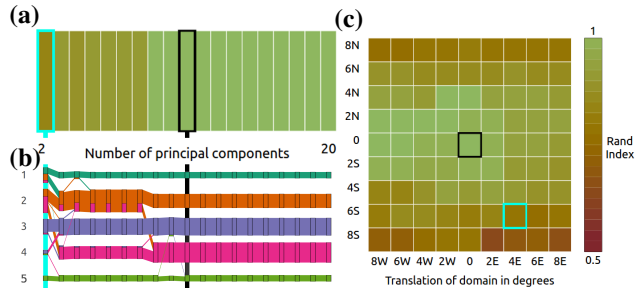

Fig. 14: Case study. **(a)** Rand index and **(b)** CSMD for changes in the number of PCs; more than 11 PCs do not change the result. **(c)** Rand matrix shows high sensitivity to translations of the domain in particular at the northern and southern edges.

For initial data inspection (Fig. 2b), we consider PV at an elevation of 250 hPa at 00:00 UTC 26 September 2016 (96 hours lead time). Fig. 12a shows the ensemble mean PV field, the ground-truth 2-PVU isocontour that was observed later, and the ensemble STDEV computed from all members. The corresponding spaghetti plot of the 2-PVU line is shown in Fig. 12b. Over the North Atlantic, the STDEV field shows high uncertainty in the vicinity and to the north of the tropopause. We are in particular interested in the formation of the ridge (the northward undulation of the 2-PVU line) to the south of Iceland and Greenland and the local tropopause depression over Southern Greenland (which is particularly hard to discern in Fig. 12b).

Based on Fig. 12a and b, we choose the large red clustering region shown in Fig. 12b, encompassing both features. To determine the number of clusters, a rather smooth curve of the elbow plot does not provide particularly strong visual guidance (not shown), we hence investigate the difference between the clusterings containing four, five, and six clusters in the corresponding CSMD and stripes plots (Fig. 13). At the transition from four to five clusters, a rather robust and substantially sized cluster 5 (light green) is formed. With six clusters, only a very small new cluster is formed (Fig. 13a). While the stripes plot shows that clusters 1 and 5 have only limited separation, STDEV reduction is further improved with five clusters compared to four clusters in the region of meteorological interest, i.e., along the tropopause in which most variability occurred (only shown for the five-cluster realization in Fig. 12c). The optimal number of PCs is found to be 11 (Fig. 14).

Interestingly, while the STDEV is reduced along the ridge, there is virtually no reduction in the vicinity of the local tropopause depression (Fig. 12c). We conclude that the ridge feature dominates the clustering but re-configure our method to use two smaller regions centered on the ridge (green region) and on the local depression (yellow region) for confirmation and to determine if we can obtain a useful clustering of the depression region. Both regions cluster best with five clusters. The green region yields slightly better separated clusters for the ridge, the STDEV reduction largely resembles that of the red region (not shown). Clustering the yellow region, however, still only leads to marginal STDEV reduction even though it is now centered on the depression (Fig. 12d). The stripes plot confirms that identified clusters are not well separated (not shown). Similarly, the CPP displaying the scenarios of the 2-PVU line shows neither well separated trends in the depression region nor in the ridge region (Fig. 12e). We conclude that data in this region cannot be well clustered.

We focus on the clustering obtained from the green region. Fig. 12f shows the corresponding CPP, showing the 2-PVU trends identified by this clustering. Cluster 2 (orange) most closely matches the observation; however, none of the cluster means represents the observation very well. The scenarios differ with respect to amplitude and orientation of the ridge. While the strongest ridge is represented by cluster 2 (orange), cluster 3 (blue) represents the cluster with the weakest ridge. In terms of ridge orientation the largest difference is found between cluster 4 (pink) and 3 (blue). Clusters 5 (light green) and 1 (dark green) are very similar to each other. This is confirmed by the CSMD and stripes plot, they are not strongly separated and cluster 5 contains a large number of members that in the four-cluster realization belong to cluster 1. We perform a robustness analysis for region translations from 2° to 8° in 2° steps in all four cardinal directions. The resulting Rand matrix and CRD are shown in Figs. 15 and 14c. While the clustering result is rather robust for east-west translations, a high sensitivity can be observed for translations in the north-south (N/S) direction. In particular for translations by more than 4° N/S the results change strongly. A closer inspection reveals,

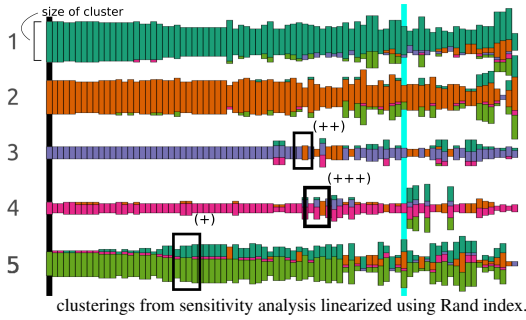

Fig. 15: Case study CRD for region translations. The green clusters (1 and 5) exchange many members (+) and in extreme cases, clusters 3 and 4 change completely (++, +++).

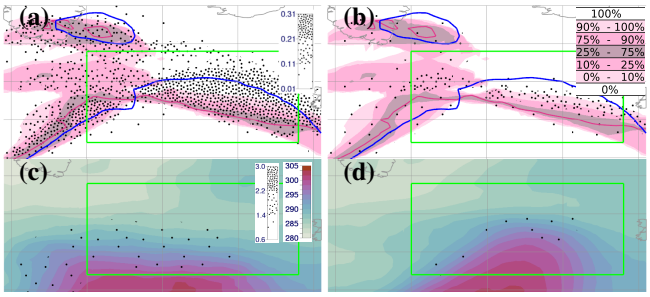

Fig. 16: **(a)** The CPP of the 2-PVU isocontour at 250 hPa for the pink cluster 4 is very sensitive to changes in the selected region (stippling shows STDEV of probability). **(b)** It becomes much more robust if the north-south region translations by 6°and 8° are omitted (cf. Fig. 14c). **(c)** Cluster means for cluster 4 and **(d)** cluster 1 (dark green) of potential temperature at 850 hPa (stippling shows STDEV of potential temperature with extreme north-south translations omitted) show distinctly different scenarios.

however, that the dominant feature (the variability associated with the strong PV gradient along the tropopause) of the cluster region is more or less removed by these translations. The high sensitivity related to these translations is thereby plausible from a physical viewpoint.

The CRD (Fig. 15) shows that members often change between clusters 1 and 5 when translating the cluster region – this is plausible due to their weak separation. The MRD (Fig. 1a) reveals the members that are switching cluster assignment. Less intuitive are exchanges between the blue and pink clusters indicated in both CRD and MRD, as the CPP shows these two clusters to represent quite different trends (Fig. 12f). Selection of the corresponding realizations in the CRD reveals that these exchanges occur for the extreme N/S translations; in these translated regions the two clusters seem to be more similar.

For further analysis, we are interested in cluster summary statistics as well as individual members that robustly remain within a given cluster. For the sake of brevity, here we only consider upper-level PV and lower-level potential temperature. The MRD (Fig. 1a) shows that for the orange cluster, the members in the upper left region of the plot appear to be robust. Inspection of further robustness details (we have selected member 26; our system also shows further members with the same robustness characteristics) reveals that these members indeed are robust with respect to E/W translations; changes occur only for N/S shifts. As a contrary example, member 50 (pink cluster) is particularly unstable (Fig. 8) and is therefore to be used with caution.

With respect to cluster summary statistics, Fig. 16 shows CPPs of the 2-PVU line at 250 hPa for cluster 4 and cluster means of potential temperature at 850 hPa for clusters 4 and 1. The variability of the respective fields due to clustering robustness is encoded as stippling. Confirming the above findings of CRD and MRD, the cluster summary statistics become much more robust when the physically unreasonable, extreme translations in the N/S directions are not considered (Fig. 16a vs. b). This increased robustness strongly increases our confidence in the subsequent physical interpretation. Clusters 1 and 4 differ with respect to the longitudinal position of the ridge maximum (cf. Fig. 12f), and also, the corresponding potential temperature fields (Fig. 16c vs. d) show distinct structure differences similar to the structure differences of the upper-level ridges. This similarity yields evidence for a baroclinic nature of the evolution, i.e., a strong coupling between the evolutions at upper and lower levels.

We conclude by noting that our preliminary analysis provides the key insight that there are robust and physically-meaningful forecast scenarios associated with the evolution of Karl in the region that was of interest during the NAWDEX campaign. Future analysis will consider in more detail the processes that lead to the divergence of the forecasts into these scenarios and the ramifications to the heavy precipitation event in Norway a few days later. Compared to methodology previously applied in the meteorological community, the method proposed here greatly facilitates the analysis and provides at the same time unprecedented quantification and visual representation of cluster robustness.

6 Summary and Conclusions

We have proposed a novel visual analysis workflow to visualize the value and, in particular, the robustness of cluster-based analyses of ensemble weather forecasts. This information is required for analysis and interpretation of the obtained clusters, however, it has been largely neglected in the context of ensemble cluster analysis in the past.

Our approach enables the user to identify a suitable number of clusters by visualizing how clusters split, are separated, and reduce an ensemble's STDEV. To investigate the robustness of a clustering, its sensitivity to changes in the selected clustering region and the number of PCs used for data reduction can be visualized. Cluster-centric and member-centric views show the stability of clusters and how members change cluster assignment when region or PCs are changed; linked visualizations of per-cluster summary statistics are augmented by overlays that indicate the sensitivity of the displayed quantities.

We have developed our methodology in a team of visualization and atmospheric scientists, integrated it into the open-source meteorological ensemble visualization software Met.3D for straightforward application by users, and have demonstrated its benefit with a real-world case study taken from ongoing work aimed at improving the accuracy of numerical weather prediction. Compared to methodology previously applied in the meteorological community, the method proposed here greatly facilitates the analysis and provides at the same time unprecedented quantification and visual representation of cluster robustness. In the near future, the method will actively be used in data analysis activities related to the NAWDEX campaign. Except for the region translations and meteorological views, we believe that the proposed workflow and the CRD and MRD can be used in other domains than meteorology as well. The extension to other deterministic clustering algorithms such as hierarchical clustering would only need small adoptions such as the use of dendrograms. Investigation of its extensibility to further clustering parameters, to time-dependent data, and to further clustering methods including fuzzy clustering would in our opinion be beneficial to the atmospheric community; it is left as an open issue for future work. A further limitation is the current brute-force computation of the clustering realization to obtain robustness information; here, approaches that can estimate this information based on fewer clustering runs would be beneficial.

Acknowledgments

The research leading to these results has been done within the subprojects B5 "Data-driven ensemble visualization" and A1 "Upscale impact of diabatic processes from convective to near-hemispheric scale" of the Transregional Collaborative Research Center SFB/TRR 165 "Waves to Weather" funded by the German Research Foundation (DFG). The work was partly funded by the European Union under the ERC Advanced Grant 291372 SaferVis: Uncertainty Visualization for Reliable Data Discovery. Access to ECMWF prediction data has been kindly provided in the context of the ECMWF special project "Support Tool for HALO Missions".

REFERENCES

[1] D. Anwender, P. A. Harr, and S. C. Jones. Predictability associated with the downstream impacts of the extratropical transition of tropical cyclones: Case studies. *Monthly Weather Review*, 136(9):3226–3247, Sept. 2008.

[2] M. Beham, W. Herzner, M. E. Gröller, and J. Kehrer. Cupid: Cluster-based exploration of geometry generators with parallel coordinates and radial trees. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1693–1702, 2014. doi: 10.1109/TVCG.2014.2346626

[3] G.-P. Bonneau, H.-C. Hege, C. Johnson, M. Oliveira, K. Potter, P. Rheingans, and T. Schultz. Overview and State-of-the-Art of uncertainty visualization. In C. D. Hansen, M. Chen, C. R. Johnson, A. E. Kaufman, and H. Hagen, eds., *Scientific Visualization*, Mathematics and Visualization, pp. 3–27. Springer London, 2014.

[4] U. D. Bordoloi, D. L. Kao, and H.-W. Shen. Visualization techniques for spatial probability density function data. *Data Science Journal*, 3:153–162, 2004. doi: 10.2481/dsj.3.153

[5] S. Bruckner and T. Möller. Isosurface similarity maps. In *Computer Graphics Forum*, vol. 29, pp. 773–782. Wiley Online Library, 2010. doi: 10.1111/j.1467-8659.2009.01689.x

[6] S. Bruckner and T. Moller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1468–1476, 2010. doi: 10.1109/tvcg.2010.190

[7] K. Bunte, S. Haase, M. Biehl, and T. Villmann. Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing*, 90:23–45, 2012. doi: 10.1016/j.neucom.2012.02.034

[8] H. Carr, D. Brian, and D. Brian. On histograms and isosurface statistics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5), 2006. doi: 10.1109/tvcg.2006.168

[9] M. J. de Hoon, S. Imoto, J. Nolan, and S. Miyano. Open source clustering software. *Bioinformatics*, 20(9):1453–1454, 2004. doi: 10.1093/bioinformatics/bth078

[10] Deutsches Zentrum für Luft- und Raumfahrt. NAWDEX - North Atlantic Waveguide and Downstream Impact Experiment. `http://www.pa.op.dlr.de/nawdex`, 2017. Accessed 30 July 2017.

[11] S. Djurcilov, K. Kim, P. Lermusiaux, and A. Pang. Visualizing scalar volumetric data with uncertainty. *Computers & Graphics*, 26(2):239–248, Apr. 2002. doi: 10.1016/s0097-8493(02)00055-9

[12] B. S. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis*. Wiley, 5 ed., Feb. 2011.

[13] L. Ferranti and S. Corti. New clustering products. *ECMWF Newsletter*, 127:6–11, 2011.

[14] F. Ferstl, K. Bürger, and R. Westermann. Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):767–776, Jan 2016. doi: 10.1109/TVCG.2015.2467204

[15] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. Visual analysis of spatial variability and global correlations in ensembles of Iso-Contours. *Computer Graphics Forum*, 35(3), 2016. doi: 10.1111/cgf.12898

[16] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann. Time-hierarchical clustering and visualization of weather forecast ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):831–840, 2017. doi: 10.1109/TVCG.2016.2598868

[17] T. H. A. Frame, M. H. P. Ambaum, S. L. Gray, and J. Methven. Ensemble prediction of transitions of the north atlantic eddy-driven jet. *Quarterly Journal of the Royal Meteorological Society*, 137(658):1288–1297, 2011. doi: 10.1002/qj.829

[18] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991. doi: 10.1002/spe.4380211102

[19] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Visualization '99. Proceedings*, pp. 43–508, 1999. doi: 10.1109/VISUAL.1999.809866

[20] L. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar. A review of robust clustering methods. *Advances in Data Analysis and Classification*, 4(2-3):89–109, June 2010. doi: 10.1007/s11634-010-0064-5

[21] J. W. Guojun Gan, Chaoqun Ma. *Data Clustering: Theory, Algorithms, and Applications*. ASA-SIAM Series on Statistics and Applied Probability. SIAM, Society for Industrial and Applied Mathematics, 2007.

[22] P. A. Harr, D. Anwender, and S. C. Jones. Predictability associated with the downstream impacts of the extratropical transition of tropical cyclones: Methodology and a case study of typhoon nabi (2005). *Monthly Weather Review*, 136(9):3205–3225, Sept. 2008.

[23] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing of extended parallel coordinates. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pp. 127–130. IEEE, 2002. doi: 10.1109/infvis.2002.1173157

[24] G. Hinton and S. Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15:833–840, 2003.

[25] T. Höllt, A. Magdy, P. Zhan, G. Chen, G. Gopalakrishnan, I. Hoteit, C. D. Hansen, and M. Hadwiger. Ovis: A framework for visual analysis of ocean forecast ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 20(8):1114–1126, 2014. doi: 10.1109/tvcg.2014.2307892

[26] M. Hummel, H. Obermaier, C. Garth, and K. I. Joy. Comparative visual analysis of lagrangian transport in CFD ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2743–2752, 2013. doi: 10.1109/tvcg.2013.141

[27] A. Inselberg. The plane with parallel coordinates. *The visual computer*, 1(2):69–91, 1985.

[28] I. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2nd ed. ed., 2010.

[29] J. Kehrer and H. Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):495–513, Mar. 2013. doi: 10.1109/tvcg.2012.110

[30] J. H. Keller, S. C. Jones, J. L. Evans, and P. A. Harr. Characteristics of the TIGGE multimodel ensemble prediction system in representing forecast variability associated with extratropical transition. *Geophysical Research Letters*, 38(12):L12802+, June 2011.

[31] F. Kovács, C. Legány, and A. Babos. Cluster validity measurement techniques. In *Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence, Budapest*, pp. 18–19. Citeseer, 2005.

[32] M. I. Krzywinski, J. E. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, 2009. doi: 10.1101/gr.092759.109

[33] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. doi: 10.1002/nav.3800020109

[34] F. Leisch. Neighborhood graphs, stripes and shadow plots for cluster visualization. *Statistics and Computing*, 20(4):457–469, 2010. doi: 10.1007/s11222-009-9137-8

[35] M. Leutbecher and T. N. Palmer. Ensemble forecasting. *Journal of Computational Physics*, 227(7):3515–3539, 2008. doi: 10.1016/j.jcp.2007.02.014

[36] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg. Comparative analysis of multidimensional, quantitative data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1027–1035, Nov 2010. doi: 10.1109/TVCG.2010.138

[37] A. L. Love, A. Pang, and D. L. Kao. Visualizing spatial multivalue data. *IEEE Computer Graphics and Applications*, 25(3):69–79, May 2005. doi: 10.1109/mcg.2005.71

[38] Y. Luo, D. Weiskopf, H. Zhang, and A. E. Kirkpatrick. Cluster visualization in parallel coordinates using curve bundles. *IEEE Transaction on Visualization and Computer Graphics*, 18, 2008.

[39] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[40] A. M. MacEachren, A. Robinson, S. Hopper, S. Gardner, R. Murray, M. Gahegan, and E. Hetzler. Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science*, 32(3):139–160, July 2005. doi: 10.1559/1523040054738936

[41] L. Magnusson, J.-R. Bidlot, S. T. K. Lang, A. Thorpe, N. Wedi, and M. Yamaguchi. Evaluation of medium-range forecasts for hurricane sandy. *Monthly Weather Review*, 142(5):1962–1981, 2014. doi: 10.1175/MWR-D-13-00228.1

[42] M. Maskey and T. S. Newman. Directional texture for visualization - new technique and application study. In *19th International Conference on Information Visualisation (iV), 2015*, pp. 1–8. IEEE, 2015.

[43] Met.3D contributors. Met.3D. `http://met3d.readthedocs.org`, 2017. Accessed 30 July 2017.

[44] T. Metsalu and J. Vilo. Clustvis: a web tool for visualizing clustering of multivariate data using principal component analysis and heatmap. *Nucleic*

Acids Research, 43(W1):W566, 2015. doi: 10.1093/nar/gkv468

[45] T. Nocke, H. Schumann, and U. Böhm. Methods for the visualization of clustered climate data. *Computational Statistics*, 19(1):75–94, Feb. 2004.

[46] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006. doi: 10.1109/TVCG.2006.170

[47] S. Oeltze, D. J. Lehmann, A. Kuhn, G. Janiga, H. Theisel, and B. Preim. Blood flow clustering and applications in virtual stenting of intracranial aneurysms. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):686–701, 2014. doi: 10.1109/tvcg.2013.2297914

[48] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, Nov. 1997.

[49] T. Pfaffelmoser, M. Mihai, and R. Westermann. Visualizing the variability of gradients in uncertain 2d scalar fields. *IEEE Transactions on Visualization and Computer Graphics*, 2013.

[50] A. Pilhöfer, A. Gribov, and A. Unwin. Comparing clusterings using Bertin's idea. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2506–2515, 2012. doi: 10.1109/TVCG.2012.207

[51] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson. Visualizing summary statistics and uncertainty. *Computer Graphics Forum*, 29(3):823–832, 2010. doi: 10.1111/j.1467-8659.2009.01677.x

[52] K. Potter, P. Rosen, and C. Johnson. From quantification to visualization: A taxonomy of uncertainty visualization approaches. In A. Dienstfrey and R. Boisvert, eds., *Uncertainty Quantification in Scientific Computing*, vol. 377 of *IFIP Advances in Information and Communication Technology*, pp. 226–249. Springer Berlin Heidelberg, 2012.

[53] L.-A. Quandt, J. H. Keller, O. Martius, and S. C. Jones. Forecast variability of the blocking system over Russia in summer 2010 and its impact on surface conditions. *Weather and Forecasting*, 32(1):61–82, 2017. doi: 10.1175/WAF-D-16-0065.1

[54] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971. doi: 10.2307/2284239

[55] Y. Rathi, S. Dambreville, and A. Tannenbaum. Statistical shape analysis using kernel pca. In *Electronic Imaging 2006*, pp. 60641B–60641B. International Society for Optics and Photonics, 2006. doi: 10.1117/12.641417

[56] M. Rautenhaus, C. M. Grams, A. Schäfler, and R. Westermann. Three-dimensional visualization of ensemble weather forecasts – Part 2: Forecasting warm conveyor belt situations for aircraft-based field campaigns. *Geoscientific Model Development*, 8(7):2355–2377, 2015.

[57] M. Rautenhaus, M. Kern, A. Schäfler, and R. Westermann. Three-dimensional visualization of ensemble weather forecasts – Part 1: The visualization tool Met.3D (version 1.0). *Geoscientific Model Development*, 8(7):2329–2353, 2015.

[58] A. Reh, C. Gusenbauer, J. Kastner, M. E. Gröller, and C. Heinzl. Mobjects–a novel method for the visualization and interactive exploration of defects in industrial xct data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2906–2915, 2013. doi: 10.1109/tvcg.2013.177

[59] D. P. Retchless and C. A. Brewer. Guidance for representing uncertainty on global temperature change maps. *International Journal of Climatology*, 36(3):1143–1159, 2016. doi: 10.1002/joc.4408

[60] M. R. Rezaee, B. P. Lelieveldt, and J. H. Reiber. A new cluster validity index for the fuzzy c-mean. *Pattern recognition letters*, 19(3):237–246, 1998. doi: 10.1016/s0167-8655(97)00168-2

[61] M. Schonlau. The clustergram: A graph for visualizing hierarchical and non-hierarchical cluster analyses. *The Stata Journal*, pp. 316–327, 2002.

[62] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

[63] R. Swinbank, M. Kyouda, P. Buchanan, L. Froude, T. M. Hamill, T. D. Hewson, J. H. Keller, M. Matsueda, J. Methven, F. Pappenberger, M. Scheuerer, H. A. Titley, L. Wilson, and M. Yamaguchi. The TIGGE project and its achievements. *Bulletin of the American Meteorological Society*, 97(1):49–67, Mar. 2015. doi: 10.1175/bams-d-13-00191.1

[64] D. M. Thomas and V. Natarajan. Multiscale symmetry detection in scalar fields by clustering contours. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2427–2436, 2014. doi: 10.1109/tvcg.2014.2346332

[65] R. L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

[66] R. D. Torn, J. S. Whitaker, P. Pegion, T. M. Hamill, and G. J. Hakim. Diagnosis of the source of gfs medium-range track errors in hurricane sandy (2012). *Monthly Weather Review*, 143(1):132–152, 2015. doi: 10.1175/MWR-D-14-00086.1

[67] X. Wang, C. Yang, and J. Zhou. Clustering aggregation by probability accumulation. *Pattern Recognition*, 42(5):668–675, 2009. doi: 10.1016/j.patcog.2008.09.013

[68] R. T. Whitaker, M. Mirzargar, and R. M. Kirby. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2713–2722, Dec 2013. doi: 10.1109/TVCG.2013.143

[69] L. Wilkinson and M. Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–184, 2009. doi: 10.1198/tas.2009.0033

[70] D. S. Wilks. *Statistical Methods in the Atmospheric Sciences*. Academic Press, 3rd ed., June 2011.

[71] J. Wu, H. Xiong, and J. Chen. Adapting the right measures for k-means clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 877–886. ACM, 2009. doi: 10.1145/1557019.1557115

[72] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008. doi: 10.1111/j.1467-8659.2008.01241.x

Visual Analysis of the Temporal Evolution of Ensemble Forecast Sensitivities

Alexander Kumpf, Marc Rautenhaus, Michael Riemer, and Rüdiger Westermann

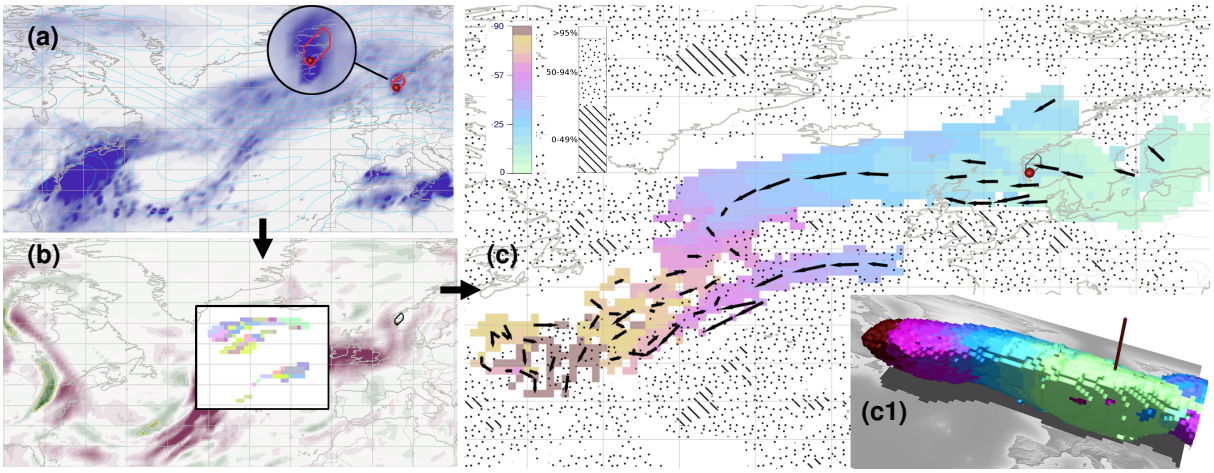

Fig. 1: Analysis of the ensemble sensitivity of forecast variable precipitation (top left) to moisture flux (bottom left) over Norway. **(a)** A high precipitation event is picked and a stochastically coherent clique is computed. **(b)** Sensitivity regions are extracted. Color shows sub-regions with high mutual correlations. **(c)** For a time sequence, sensitivity regions are matched between time steps and tracked over time. A "swipe-path" colors (time in h) all locations covered by a selected sensitivity region over time according to the first time of coverage. Stippling covers statistically insignificant regions. **(c1)** The proposed workflow operates on 2D and 3D data.

Abstract—Ensemble sensitivity analysis (ESA) has been established in the atmospheric sciences as a correlation-based approach to determine the sensitivity of a scalar forecast quantity computed by a numerical weather prediction model to changes in another model variable at a different model state. Its applications include determining the origin of forecast errors and placing targeted observations to improve future forecasts. We—a team of visualization scientists and meteorologists—present a visual analysis framework to improve upon current practice of ESA. We support the user in selecting regions to compute a meaningful target forecast quantity by embedding correlation-based grid-point clustering to obtain statistically coherent regions. The evolution of sensitivity features computed via ESA are then traced through time, by integrating a quantitative measure of feature matching into optical-flow-based feature assignment, and displayed by means of a swipe-path showing the geo-spatial evolution of the sensitivities. Visualization of the internal correlation structure of computed features guides the user towards those features robustly predicting a certain weather event. We demonstrate the use of our method by application to real-world 2D and 3D cases that occurred during the 2016 NAWDEX field campaign, showing the interactive generation of hypothesis chains to explore how atmospheric processes sensitive to each other are interrelated.

Index Terms—Correlation, clustering, tracking, ensemble visualization.

1 Introduction

Ensemble sensitivity analysis (ESA) has been established in meteorology as an ensemble-based approach to estimate the sensitivity of a scalar forecast quantity J (in meteorology referred to as "forecast metric") computed from numerical weather prediction (NWP) model output to changes in the same or another model variable at an earlier state. ESA has been introduced by Ancell and Hakim [1] and Torn and Hakim [42]. It is a correlation-based approach which considers the variations of physical quantities at different domain locations *relative* to each other.

ESA has been used in a number of applications, including identification of the origins of forecast errors (e.g., [23]), investigation about relations between specific atmospheric processes to other processes of interest in an NWP model (e.g., [43]), and planning of "targeted observations" aiming at improving future forecasts by increasing observation density in critical regions of the atmosphere (e.g., [24]). The use of ESA, however, is associated with a number of challenges.

One major challenge (C1) is how to assess the confidence that can be put into the computed sensitivities. The metric J is commonly obtained by manually selecting a spatial region in which a certain weather event has been identified, and by using a single representative measure for the entire region (e.g., mean or root-mean-squared error of corresponding meteorological parameter) to compute J for every ensemble member. This approach can become a major source of uncertainty, since a single measure does not consider the distribution of values over a region and, thus, might not sufficiently capture the event of interest in all members.

- *Alexander Kumpf, Marc Rautenhaus*, Rüdiger Westermann are with the Computer Graphics & Visualization Group, Technische Universität München, Garching, Germany. *M. R. is now with Universität Hamburg, Regional Computing Center, Hamburg, Germany. E-mail: {alexander.kumpf, marc.rautenhaus, westermann}@tum.de*
- *Michael Riemer is with the Institute of Atmospheric Physics, Johannes Gutenberg Universität Mainz, Mainz, Germany. Email: mriemer@uni-mainz.de*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

Furthermore, the stochastic stability of the event in the selected region, i.e., the likelihood of variations in the predictions relative to each other, cannot be inferred from a single measure. Thus, the chance of occurrence of the selected event might be very low.

Once a metric J has been defined, ESA computes the correlation between J (the target parameter) and the ensemble data of another model variable (the source parameter) at every point in the domain (cf. [42]). In the resulting correlation field sensitive regions can be identified. Then, another challenge (C2) is to infer the geographical origin of forecast errors from the temporal evolution of theses sensitivities (e.g., [23, 47]). Since the manual tracking of sensitive regions (e.g., [47]) becomes tedious especially if many parameters are of interest, addressing C2 entails the development of automatic approaches for tracking sensitive regions in a reliable way. Furthermore, it needs to be analyzed whether determined regions exhibit sufficient stochastic stability (C3) regarding the variation of model parameters across these regions. If this is not the case, it is very likely that the corresponding atmospheric structure will fall apart over time, so that conclusions drawn from ESA have to be carefully evaluated.

Another challenge (C4) put forward by our collaborators from meteorology was the application of ESA to analyze sensitivities in three-dimensional space. In existing meteorological workflows ESA is solely applied to analyze the sensitivity between field variables on two-dimensional atmospheric levels (e.g., [23, 43]). To the best of our knowledge, there is no study that applied ESA in full three-dimensional space, even though such an extension would be very beneficial to investigate the interrelations of atmospheric processes as well as the structure of related weather events in the inherently three-dimensional atmosphere.

1.1 Contribution

We introduce a novel visual analysis workflow for addressing C1 to C4; an overview of this workflow is provided in Sect. 3 and Fig. 2. We demonstrate its application by analyzing weather systems observed during the 2016 North Atlantic Waveguide and Downstream Impact Experiment (NAWDEX, [37]), an atmospheric research field campaign in which two of the authors were involved. The NAWDEX research on predictability of weather investigates how different physical processes can play different roles in cyclone evolution; our presented ESA approach aims at facilitating insights into sensitivities of the weather evolution related to Tropical Cyclone "Karl" in September 2016.

To address C1, we provide options to determine sub-regions with low stochastic variation in the region in which the selected weather event occurs. Therefore, the user can pick a location at which the event has a significant occurrence, and via correlation clustering—adapted from Pfaffelmoser and Westermann [27]—the set of locations (including the picked one) with mutual correlations above a threshold is determined. A region comprised of such locations is said to be coherent with respect to the predicted event, and we call such a region a *correlation clique*. By repeating this process, multiple coherent regions can be determined, and the most representative one can be selected and used to restrict the computation of J to its locations.

We address C2 by letting the user, in the source parameter field, interactively select connected components of superlevel sets, i.e., the set of locations at which a threshold in the normalized sensitivity field is exceeded, and track these components over time. Furthermore, and to address C3, we provide an indication of the robustness of the ESA result by visualizing correlation clusters in a selected superlevel set. Computed clusters indicate low or high stochastic variation of model parameter values across the selected region, letting the user infer the structural coherence in the corresponding weather event.

To track a selected feature, i.e., a superlevel set, we use an area-weighted optical flow based approach. To match features in two successive time steps, for all locations covered by a feature in the current time step, we vote using an uncertainty-aware correspondence measure whether these locations again belong to a feature in the next time step.

We account for uncertainties in a feature's boundary by assigning target locations close to a boundary, yet outside a feature, to the closest location inside that feature so that correspondence is established. Additionally, since the features we consider can vary significantly in size over time, the optical flow is used in forward and backward direction in combination with correspondence statistics to obtain a measure that better takes into account the size of matched features.

To effectively convey sensitivities between events at different atmospheric levels as well as the temporal changes these events undergo (C4), we have designed a system comprised of two linked views: An abstract view displaying a graph structure showing split and merge events, and a single track visualization using a so called swipe-path in multi-dimensional space. These visualizations enable the user to shed light on the robustness of tracked events over time as well as their temporal changes in shape, size, and location at different atmospheric levels.

2 Related work

ESA has been introduced by Ancell and Hakim [1] and Torn and Hakim [42]. It has been used in various meteorological publications with a prominent use case being the estimation of the impact of additional observations on forecast accuracy [42]. Although alternative sensitivity analysis techniques are available and used in meteorological applications (an overview is provided e.g., by [24]), due to the widespread use of ESA, and in particular its use by our meteorological partners, we focus exclusively on this method in the current work.

With respect to visualization research, our approach is related to techniques for ensemble visualization, and in particular correlation visualization and region-based tracking in time-varying multi-dimensional scalar fields. A thorough overview of ensemble visualization techniques in meteorology can be found in the recent survey article by Rautenhaus et al. [29].

Finding a suitable visual representation of non-local statistical quantities such as correlation structures is one of the most challenging tasks in visualization. Visualizing the correlation matrix directly is unfeasible for large datasets and cannot show spatial relationships. Global correlation structures were determined by Bansal et al. [3] via correlation clustering, which groups objects based on pair-wise similarities (positive correlation) and dissimilarities (negative correlation) using graph partitioning. Liebmann et al. [21] clustered correlations based on distances on hyperspheres. Paffelmoser and Westermann [27] introduced correlation-based region growing to determine clusters in which the degree of dependency between the data at the cluster centroid and the cluster locations does not fall below a threshold. In contrast, in our approach we search for clusters of locations for which the *mutual* data correlations are above a threshold, to ensure that the stochastic stability of the entire cluster is high.

Paffelmoser and Westermann [28] derived a model to represent local anisotropic correlation structures and used this model to distinguish between correlations along and orthogonal to isosurfaces in 3D scalar fields. By using this approach they analyzed the possible variations of isosurface structures in uncertain scalar fields. Other approaches restricted the analysis to the correlations between the data values at the *same* location in different datasets [16, 36], or they analyzed the data variations at the same locations over time, often via variants of time-activity curves [9, 40]. Chen et al. [6] used a sampling-based approach to summarize temporal correlations between voxels in multivariable and time-varying datasets with 3D spatial references. Recently, Zhang et al. [46] developed a temporal multi-variable structure that can express temporal information at a location in multi-dimensional space. This was combined with a dissimilarity-preserving cluster algorithm that characterizes time-varying patterns and spatial locations.

In visualization, a number of techniques have been developed to track regions over time, i.e., to establish the correspondence between regions from one time step to the next. In its simplest form, such methods track the connected components of regions where the data values are entirely below or above a given threshold [35, 38], yet also more sophisticated global feature analysis based on scalar field topology and statistics has been used [2, 5, 7, 34, 39, 45].

The correspondence between regions can either be found using overlap calculations [35, 38] or by matching attributes that describe specific properties of each region [31, 33, 35]. When data with high

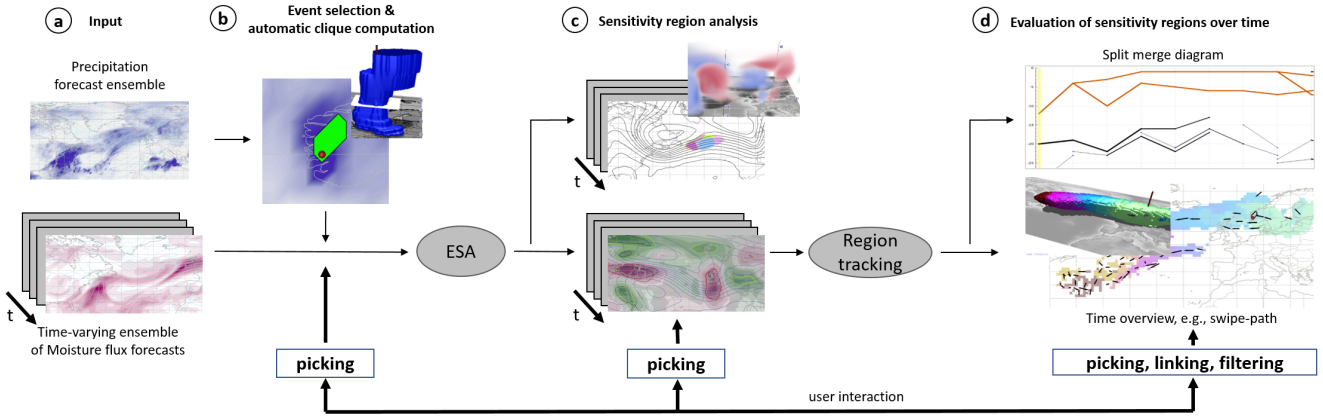

Fig. 2: Method overview: (a) Input are two ensembles of 2D (3D) scalar fields, one of them resolved over time. (b) In the input field of parameter J, a point of interest is picked. A stochastically coherent clique around this point is computed. ESA is then used to compute sensitivity values at every domain point to this clique. (c) Sensitivity fields are displayed, and selected regions can be further analyzed by means of correlation clustering. Selected regions are tracked over time, and (d) different visualizations like split-merge diagrams and swipe-paths are computed and used to analyze sensitivity tracks.

temporal resolution is given, region overlap methods require only a matching function which measures the degree of similarity of regions in different time steps. In early work this was done using area/volume of the overlap region, yet more advanced matching functions using the Earth Mover distance [17] or the Jaccard distance as isosurface similarity metric have been employed recently [12, 34]. However, since these metrics do not consider the size of the overlap regions relative to the features' spatial extent, the matching result can be misleading; for instance, if a large feature splits into multiple very small features.

When data with low temporal resolution is given and features vary strongly in shape and position over time, as it commonly occurs in our meteorological use cases, overlap-based approaches face an additional problem: Before analyzing the overlap, the direction into which the feature has moved needs to be predicted. Muelder and Ma [26] addressed this problem by predicting a feature's next location from previous locations using trajectory extrapolation, and then computing the overlap between the shifted and a target feature. Doraiswamy et al. [7] and Valsangkar et al. [2] used the optical flow [15, 22] between brightness temperature and pressure fields, respectively, to obtain an initial guess of the location of a cloud or cyclone feature in the next time step. Two features are matched if there is at least one hit between these features, i.e., a location in the region of the source feature is connected to a location in the region of the target feature.

In meteorological and climate research tracking is often reduced to single points like extremum points in pressure or precipitation fields [14], and these points are then connected together across time to form tracks. A popular method to find corresponding points in successive time steps is the first-guess method [44], which predicts the location of a selected point in the next time steps by linear continuation and then searches for the closest feature point in the next time step. Gambheer and Bhat [13] tracked clouds by considering the overlap of closed feature contours across time steps. More recently, Fiolleau and Roca [11] considered the time-varying two-dimensional input as a three-dimensional volume, and tracked clouds by tracking seed points within this volume.

3 Method Overview

Given an ensemble of forecast runs, our proposed visual analysis workflow enables meteorologists to interactively analyze the sensitivity of a selected model parameter, e.g., precipitation, to another one, e.g., moisture flux. Fig. 2 shows an overview of the proposed workflow. We consider scalar field data from the ECMWF Ensemble Prediction System (ENS; e.g., [20]). The ensemble comprises 50 perturbed members and an unperturbed control forecast (that is started from the "best" initial conditions). Past forecast data is available from the TIGGE archive [41] on a regular longitude–latitude grid in the horizontal; in the vertical, data is available on levels of constant pressure.

The user first inspects the dependent parameter for which the sensitivity shall be analyzed. Following common practice in meteorology, we show a 2D map of ensemble values, e.g., the point-wise mean or maximum values (Fig. 2a). In workflows from past meteorological publications (e.g., [24, 43]), users select a target region enclosing a significant weather event for which ESA is performed. The parameter values in the selected region are then used to compute the single value J per ensemble member, e.g., a mean value or a root-mean-squared error. Then, at every domain location the sensitivity of these values to the ensemble values of the independent parameter at these locations are computed.

It is here were we place our first improvement of the workflow: We let the user determine a representative region in which an event is coherently predicted, i.e., a correlation clique (cf. Sec. 4.1). Therefore, the user selects a seed location, and the system computes instantly a correlation clique to this location (Fig. 2b, Fig. 3). The size of this clique and its strength, i.e., the strength of the mutual dependencies between the data values at the locations covered by the clique, indicate the expected reliability of the ESA results (C1). The user can interactively select different seed locations to find the largest or strongest clique, or abort the process if no such clique is found. For computing the metric J, the parameter values are then condensed only over the selected clique.

The sensitivity values computed by ESA (cf. Sec. 4.1) are plotted in a separate map (Fig. 2c). To highlight regions which can be deemed a significant influence on the target region, the superlevel sets with respect to a sensitivity threshold—selected based on a significance test—are visualized. To address C2, the user selects a region, and the system automatically tracks this region over time by matching against regions in successive time steps (Fig. 2d). Prior to tracking, however, our second improvement comes into play to address C3. By means of correlation clustering it is indicated whether the parameters in the selected superlevel set show low or high stochastic variation (Fig. 2d). If, for instance, two large clusters with inverse correlation to each other exist, the corresponding weather event might be unstable and, thus, rather unsuited for a temporal evaluation.

The tracking process (cf. Sec. 4.3) is guided by the optical flow to predict the percentage of area of one region that overlaps with another region in the next time step. By using the optical flow forward and backward in time, we detect split and merge events, which are encoded in an abstract split-and-merge tree (Fig. 2d) that is linked to the spatial view. This allows for an interactive inspection of detected events, by picking objects in the tree and visualizing them in their spatial

surrounding. For a selected region we compute a so-called swipe-path (Fig. 2d), to reveal the spatial and geometric variations a region undergoes over time. A swipe-path shows in one image the temporal evolution of a region, including all split and merge events by coloring and blending all single time states together (Sec. 4.3.2).

To address C4, all components of our proposed visual analysis workflow can be performed optionally using 2D or 3D data (see inlets in Fig 2). To ease navigation, region selection is always performed on 2D maps, and in a synchronized view the 3D region is shown simultaneously. When 3D data is used, direct volume and isosurface rendering via GPU raycasting is used to visualize extracted regions and show their tracks over time.

4 Ensemble Sensitivity Analysis

We introduce details of our proposed interactive visual analysis workflow for using ESA, and we demonstrate its outcome and derived hypotheses on a real-world dataset. We elaborate specifically on the additional indications and suggestions that our workflow offers compared to the common use of ESA in meteorology.

4.1 Statistically coherent input regions

The ESA approach diagnoses statistically the sensitivity of a selected forecast quantity (the input metric) for a target location to another quantity at other locations and prior times (the initial condition state variable). ESA determines these statistical relations using correlation measures. In particular, the sensitivity $(\partial J/\partial s)$ of the target forecast metric J to a selected state variable s is computed at all points in the model domain. This sensitivity relationship can be expressed as

$$\frac{\partial J}{\partial s} = \frac{\mathrm{cov}(J,s)}{\sigma(s)}, \quad \text{with} \quad \mathrm{cov}(J,s) = \frac{1}{n-1}\sum_{k=1}^{n}(J_k-\bar{J})(s_k-\bar{s}).$$

Here J and s are n-dimensional data vectors, with n being the ensemble size, and the covariance (cov) and standard deviation (σ) are computed over all ensemble members. Thus, the sensitivity measure is the Pearson correlation coefficient multiplied by the standard deviation of the state variable $\sigma(s)$ at a certain location.

As the data values in regions exhibiting very low correlation can be assumed independent of each other, the effect of uncertainty on a feature in such a region is to a large extent arbitrary. Since $\sigma(s)$ can be seen as a measure of the ensemble uncertainty in the state variable at a certain location, points with both high correlation and high uncertainty are emphasized, causing weaker structures to be filtered out. The underlying rational is that changes in terms of standard deviations to grid points with small uncertainty will influence the outcome of a weather prediction system only slightly.

Using the ESA measure, so called hypothesis chains between physical events can be generated to identify physical processes that potentially cause uncertainty in the forecast. The combination of meteorological fields for the input metric and state variable (e.g., precipitation and moisture flux) generates the ensemble that is analyzed to quantify the contribution to uncertainty induced by the state variable with respect to the natural variability estimated from the meteorological models. Critical to the sensitivity analysis is the selection of a suitable input region over which the metric J is considered.

Using the input metric ensemble values at a single location and computing sensitivities of these values to the state variable ensemble values at other locations is highly sensitive to the chosen location. Hence, in practice a region that contains a significant weather event, i.e., a meteorological process of interest like heavy precipitation, is selected, and for each ensemble member a single measure for the entire region is computed and used as input metric J. Also more sophisticated selection procedures exist, for instance, which split manually the selected region into coherent sub-regions with respect to orography or texture, yet all these approaches require profound meteorological knowledge to select a meaningful set of input locations. Furthermore, as indicated by Fig. 4, computed sensitivities are highly sensitive to the selected input region.

We determine automatically a statistically coherent set of locations over which the input metric is computed, by computing *correlation cliques*. A correlation clique is defined as the set of locations (including a user-selected seed location) with pair-wise correlations between the data values at these locations above a threshold. A clique can consist of multiple connected components. The input metric J is then computed ensemble-member-wise over these locations, and the sensitivity is computed to the resulting ensemble of values of J. In this way, only locations that show a statistically similar data distribution and that are likely to deviate into the same direction in data space are considered. Averaging over regions with completely different statistical characteristics is avoided. Fig. 3 shows correlation cliques for 5 different seed locations that were selected in the depicted precipitation field. It can be seen how strongly the cliques can vary, and how few locations of the high precipitation event they can cover.

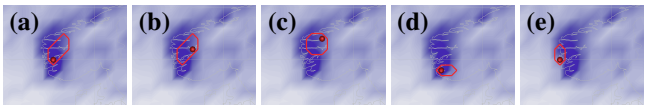

Fig. 3: 24 hours ensemble maximum precipitation over Norway, computed between 27 - 28 September 2016, 18:00 UTC. Comparison of correlation cliques computed from different seed locations.

Figure 4 shows sensitivities computed by ESA for J computed over two manually selected rectangular regions and a correlation clique. Since the orography over Norway is vastly different for every grid-point, it is difficult to select a rectangular region covering a coherent region. The use of an automatically adapted correlation clique can alleviate this problem.

4.1.1 Computation of correlation cliques

To compute a correlation clique, the user first selects a domain location (seed location) at which a significant weather event is predicted, e.g., a location with high precipitation. Given this location, the clique is computed via Algorithm 1.

input : X, $\hat{x}$, α
$x_c = \hat{x}$; $\tilde{X} = X$; $i = 0$;
sort $\tilde{X}$ w.r.t. decreasing $\mathrm{corr}(s(x_j), s(x_c))$, $x_j \in \tilde{X}$;
while $i < |\tilde{X}|$ **do**
  $\tilde{X} = \tilde{X} \backslash \{x_j | \mathrm{corr}(s(x_c), s(x_j)) < \alpha,\ x_j \in \tilde{X}\}$;
  $x_c = x_{i+1}$;
  $i = i + 1$;
end
return $\tilde{X}$;

Algorithm 1: Our algorithm for computing a correlation clique, with X containing all locations (i.e., grid points) in the domain, $\hat{x}$ the seed location, x_i the i^{th} element in $\tilde{X}$, $s(x_i)$ the value vector at location x_i, and α the selected correlation threshold.

To perform the computation efficiently, we first determine all locations with a correlation to the seed location that is greater than or equal to the threshold, sort these locations with respect to decreasing correlation to the seed location, store the sorted sequence, and proceed as follows: As long as there are locations in the sequence, we select the one with highest correlation to the seed location, and remove all locations from the sequence with a correlation to the selected one below the selected threshold. This procedure is successively applied until all locations have been processed, and the remaining locations belong to the computed correlation clique. The correlation threshold has a significant effect on the size of the cliques; the lower (higher) it is, the more (less) locations will be assigned to a clique. From a number of experiments we have found empirically that a threshold around 0.7 usually leads to plausible cliques. In particular, we have observed that slight variations around 0.7 often do not cause any significant changes, and that resulting cliques were considered representative for the considered event by our partners from meteorology.

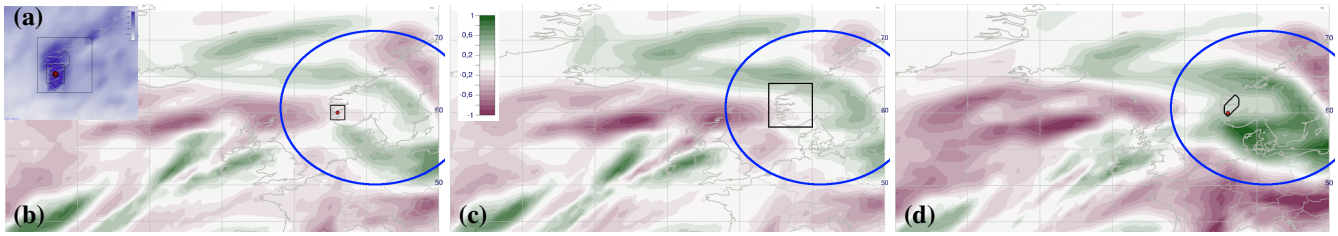

Fig. 4: **(a)** As in Fig. 3, together with the different input regions (framed in black) that were used to compute the ESA metric J. **(b)**-**(d)** Sensitivities are computed with respect to moisture flux at 18:00 UTC 28 September 2016, corresponding to the time right after a high precipitation event over Norway. Hence, high sensitivities are expected mostly east (downstream, blue circle) of the high precipitation region. Rectangular input regions **(b)**, **(c)** show weaker signals than the region that was automatically determined via a correlation clique **(d)**, especially east of the event.

Upon computing a correlation clique, it is displayed as an isocontour in the scalar field of input metric J. Locations, i.e., grid points, belonging to the clique are set to one, while all others are set to zero, and the isocontours are extracted from the resulting binary field to the threshold 0.5. The same procedure is applied in 3D, yet the cliques are displayed as isosurfaces in the 3D domain.

4.2 Coherent sensitivity regions

With a selected correlation clique, at every domain point ESA is used to compute the sensitivity between the state variable s at that point and the input metric J. Of special interest are regions of high sensitivity, i.e., superlevel sets to a selected sensitivity threshold, displayed by their enclosing level set. These regions are considered to be coherent with respect to sensitivity, and they are used to indicate where the prediction of J can be improved, i.e., by assimilating additional data from measurements. However, it can be conjectured that measurements placed in a region with high statistical variance will not be able to improve the prediction. Instead, a separate measurement for each statistically coherent region should be preferred.

To further shed light on the statistical variance within a selected region of high sensitivity, one possible solution is to show the correlation matrix for all locations belonging to that region. In general, however, this is unfeasible because the correlation matrix is too large and structures between the values at different locations cannot easily be identified from it. To overcome this limitation, we utilize the algorithm for computing correlation cliques to partition the region into multiple disjoint cliques, with all positive correlations between the data values at locations in one clique.

4.2.1 Correlation clustering

Algorithm 2 describes the procedure for computing a set of correlation cliques that densely cover a selected superlevel set. Our goal is to determine regions with low internal yet high mutual stochastic variation. Multiple cliques with low mutual stochastic variation indicate a rather unstable weather event that may fall apart over time, hinting towards regions where the meteorological models tend to produce different predictions. Especially sub-regions with very low or even negative correlations to the initial clique should be treated separately in subsequent analyses.

The algorithm first computes a correlation clique in the field of state variable s to the seed location with highest sensitivity to J. Next, a second clique is computed, that also has high correlations between the data values at assigned locations, but with low correlation to the first clique. Therefore, we compute the member-wise means of the state variable over the initial clique and select from the remaining locations the one with the minimum correlation to the mean values as seed point for the second clique. This process is then repeated with the second clique as initial clique until all locations in the superlevel set have been assigned.

The resulting partition shows the size and location of cliques with strongly correlated data values, yet with positive and negative mutual correlations. For different scenarios this is demonstrated in Fig. 5. Even though inverse correlations between cliques within one single region with high sensitivity seem surprising at first, such structures can nevertheless occur because the Pearson's correlation coefficient underlying ESA is not transitive.

input : X, $\hat{x}$, α
$\tilde{X} = X$; $i = 0$;
$\text{clique}_0 = \text{correlation\_clique}(\tilde{X}, \hat{x}, \alpha)$;
$\tilde{X} = \tilde{X} \backslash \{x | x \in \text{clique}_0\}$;
while $(\tilde{X} != \emptyset)$ **do**
  $s_c = \text{compute\_metric}(\text{clique}_i)$;
  $x_i = \arg\min_{x \in \tilde{X}}(\text{corr}(s(x), s_c))$;
  $\text{clique}_{i+1} = \text{correlation\_clique}(\tilde{X}, x_i, \alpha)$;
  $\tilde{X} = \tilde{X} \backslash \{x | x \in \text{clique}_{i+1}\}$;
  $i = i + 1$;
end

Algorithm 2: Correlation clustering splits a selected region X into sub-regions with high interior correlation but low mutual correlation.

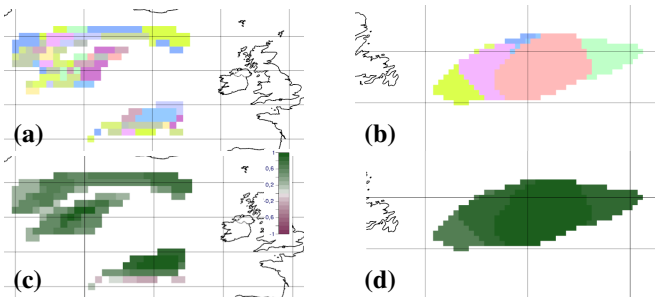

Fig. 5: Partition of superlevel sets using correlation clustering (Alg. 2). Level set in ESA between **(a)** precipitation and moisture flux and **(b)** geopotential height error and geopotential height. Many small clusters (different colors) in **(a)** indicate high stochastic variability. Large clusters **(b)** indicate large statistically coherent regions. **(c,d)** Correlation of first cluster to all others in the region. Even negative correlations (red) appear.

4.2.2 Significance of sensitivity

To test whether an extracted region of high sensitivity is significant, a two-tailed t-test is commonly applied to the correlation values in a selected region ([47]). For the 51 member ensembles used in our work, values higher than 0.276 can be deemed significant with 95% confidence. This, however, can only be seen as a rough indicator, since for every fixed J the t-test assumes normally distributed values for s, which cannot be guaranteed in general. Therefore, we use the test solely to remove regions with low significance. Once a region has significant parts, it is considered as a whole in the upcoming analysis. In our visualizations, we use an additional visual channel to show the significance of the sensitivity values over the domain. According to Retchless and Brewer [32] we use stipple patterns as overlays to

depict the significance. In particular, we use a point stipple pattern with constant point density to indicate regions with medium significance, i.e., between 50% and 94%, and a line stipple pattern with constant line spacing to indicate very low significance, i.e., below 50%. The overlays are generated by mapping textures filled with the stipple patterns over the entire domain, and fading out a texture's contribution where the significance values are not in the corresponding interval. The different types of stipple overlays we use are demonstrated in Fig. 1c.

4.3 Tracking sensitive regions

In the majority of meteorological workflows for ensemble analysis today, tracking of regions in which significant weather events are predicted is performed via animation and manual matching of corresponding regions in successive time steps. One of the reasons is that the predictability of regions associated with high-impact weather events, which are often the regions of interest, is low, and predicted events tend to undergo major changes in shape and location over lead time. To support the automatic tracking of regions over time also in this situation, we propose an improved tracking algorithm using the optical flow in forward and backward direction.

4.3.1 Bidirectional OF-based matching

Given a sequence of 2D or 3D time-varying scalar fields, e.g., precipitation, the optical flow (OF) [15, 22] estimates the apparent motion of precipitation patterns in two successive fields, the source and the target field. The main idea behind the OF algorithm is to minimize a global cost function that represents the rate of change of the scalar quantity from the source to the target, under the assumption that the metric does not change during a sufficiently small duration. The result is a displacement field, which indicates for every spatial location and given data value at that location to which location this value should be moved such that the transformed field matches the target field. In our workflow we use the implementation of the Farneback algorithm [10] that is provided by OpenCV [4], with a window size of 12 in either dimension, pyramid scale of 0.4, polynomials of degree 8, 2 iterations, and a smoothing factor of 1.2.

To establish a correspondence between regions in the ESA field at different time steps, which are defined as superlevel sets to the correlation threshold, we apply forward *and* backward OF between these fields (see Fig. 6). In the forward pass, the OF field is used to

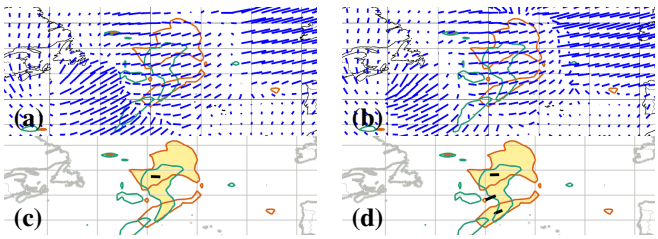

Fig. 6: Forward **(a)** and backward **(b)** OF between sensitivity fields at two consecutive time steps. **(c)** Forward OF tracking with 25% filtering matches a single region (orange outline, yellow interior) to the picked region (green outline, yellow interior). **(d)** Bidirectional OF tracking using the same filtering matches two additional regions, one of them a major split event visible in Fig. 1c.

estimate at which location in the local surrounding of every source location a similar ESA value is found in the next time step. This leads to the predicted motion of the ESA field over the corresponding time period. In principle, this motion field can already be used to establish a correspondence between regions in successive time steps: Correspondence between a source and target region is established if there is at least one location in the source region where the motion vector points at a location in the target region [2]. As manually validated by our domain experts, however, this approach has led to a number of falsely matched regions as well as missed connections in our specific scenario.

To weaken this limitation, we have modified the OF-based matching procedure in two ways: Firstly, for every target location that is indicated by the motion vector, we test whether this location is sufficiently close to a target region. This indicates that the location belongs to that region with high probability. In particular, we test for a distance less than the size of one cell of the grid structure at which the ESA values have been computed. The total number of matches from the source to the target region, as well as the percentage of points of the source region that are matched to a certain target region is stored. The number of matches is used in the split-merge diagram described below to indicate the size of matched regions, and the percentage of matched locations is used to filter out region pairs that are only connected weakly.

Secondly, we use the OF, now oriented backward in time, to estimate from where in the source field a physical quantity was moved over the current time step. We could do this via a semi-lagrangian advection step known from fluid simulation, where the motion vectors in the target field are simply reversed, yet the explicit backward step using OF enables us to employ a bidirectional matching according to the distribution of the values in both fields. The result of the backward step is used in the following way: Firstly, we establish a new link in exactly the same way as in the forward step, and we store the number and percentage of matched locations. The computed links are now filtered with respect to a prescribed percentage threshold: If for a given pair of matched regions in both the forward and the backward step this threshold could not be reached, the link is disconnected. Fig. 7 illustrates these two scenarios. Our partners from meteorology liked in particular the possibility to interactively change the percentage threshold and see immediately the resulting matches in the spatial view (see below). By this, it was easy for them to explore all suggested pathways of weather events and select those in best agreement with their domain-specific assumptions.

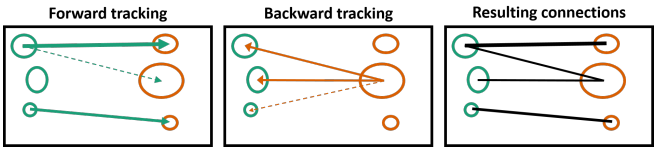

Fig. 7: Schematic illustration of bidirectional OF-based region tracking. Matches (indicated by arrows) identified between regions in time step t_i (green circles) and t_{i+1} (orange circles) are shown for forward and backward OF tracking. Arrow width indicates number of matched locations. Dashed lines indicate weak matches which are filtered out. Final result is the union of remaining matches.

4.3.2 Visual encoding of region tracks

The computed matchings between source and target regions over the entire simulation time are displayed in a split-merge diagram, where every sensitivity region is represented as a node and edges between nodes are established if a match has been established (Fig. 8b). The number and percentage of matched locations for a selected region is used to adjust the width of edges in this graph (Fig. 8b). The diagram orders all extracted regions along the horizontal time axis, yet the regions in one single time step are drawn along the vertical axis in an unordered way. It is clear that this can lead to significant changes in the edge orientations from time step to time step. However, since the split-merge diagram is solely used as an additional support tool for selecting a region that can then be tracked in the spatial view, we did not focus on improving its visualization. Improved layouting strategies for such diagrams are discussed by Widanagamaachchi et al. [45]. In the split-merge diagram time decreases along the horizontal axis. The resulting unusual ordering of time steps was specifically requested by our collaboration partners from meteorology. Since relevant sensitivity structures are defined close to the selected event and then traced backward in time, our partners wanted to see the same temporal evolution in the diagram.

In the diagram the user can now pick a time step and let all extracted

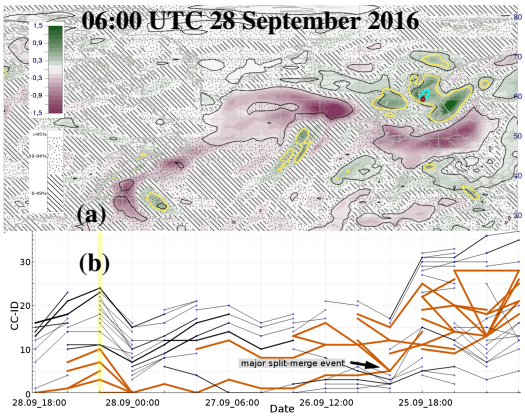

Fig. 8: **(a)** ESA sensitivity between precipitation and moisture flux (color). Stippling covers regions with less than 95% significance. The region with positive ESA close to the user-placed pole (red dot) is selected for tracking, which is performed towards the initial time of the simulation with increasing time-difference to the precipitation event. **(b)** The split-merge graph with 10% filtering shows that the selected region can be traced back (orange path) to the initial time step. There is a merge event which connects two major tracks (black arrow).

regions in that time step being displayed in the linked spatial view. Furthermore, we let the user select a region in the spatial view, and show all regions that were merged with this region or which emerged due to a split of this region. From this set of regions, i.e., the connected component C in the diagram containing the selected region, a so-called swipe-path is generated in the spatial view. The swipe-path encodes the spatio-temporal evolution of the selected region, by assigning to every location that was overlaid by a region in C a scalar value indicating the first time this location was covered. In this way, the evolution of a region is encoded in a scalar field, which can then be visualized by isocontouring, 2D color plots, or direct volume rendering in 3D (see Figs. 11c and 1c). We do not smooth the resulting spatial structure to keep the grid resolution visible, which was also requested by our partners from meteorology. When mapping scalar values to colors, gradually changing colormaps are beneficial since neighboring time steps can be identified while the whole evolution remains clearly visible.

Even though a swipe-path, conceptually, is a rather simple visual representation of the temporal evolution of selected regions, especially its use for depicting the motion of 3D regions was very well received by our partners from meteorology. For the first time ever, meteorologists could investigate the vertical movement of sensitivity regions, including the geometric changes these regions undergo over time. The domain experts argued explicitly against the use of glyphs for depicting the temporal evolution of a selected region, to be able to directly read both the spatial and temporal changes a region undergoes.

5 Results and Evaluation

All components of the proposed visual workflow have been integrated into the open-source meteorological visualization tool "Met.3D" [25, 30]. The existing data processing pipeline of Met.3D, as well as existing visualization functionality for meteorological maps and direct volume rendering, provided a suitable pre-existing infrastructure. Also, integration into an existing tool eases promulgation of our approach into the meteorological community.

Firstly, we demonstrate the use of the workflow to investigate sensitivities in weather forecasts related to the extratropical transition (ET) of tropical cyclone "Karl" in late September 2016. The case is a focus of current NAWDEX analyses, aiming at identifying atmospheric processes that may have caused deficiencies in the predictability of the subsequent weather evolution (cf. [37]). In the days considered, Karl moved into the middle latitudes, merged with a pre-existing weak extratropical cyclone, rapidly re-intensified during this extratropical transition and thereby impacted the jet stream over the North Atlantic and northern Europe. This period was characterized by large forecast errors. Visualization methods to investigate further aspects of the case were already presented by Kumpf et al. [19] and Kern et al. [18]. We use data from the ECMWF ensemble prediction system (ENS; e.g., [20]); all results are produced from data on a regular latitude–longitude grid with a grid spacing of 0.5°(1° for precipitation), using levels of constant pressure in the vertical.

We consider an extreme precipitation event that occurred along the Norwegian coast at the end of Karl's life cycle on 28-29 September 2016. In some places, more than 116 mm of rain fell in less than 24 hours. A large-scale and a smaller-scale perspective are analyzed, yielding first insights into the atmospheric processes involved.

5.1 Large-scale perspective: Geopotential height error

Fig. 9 shows the forecast error (the difference between forecast and subsequently observed values) of 300 hPa geopotential height (gravity-adjusted height above sea level) of the ensemble control forecast at 00:00 UTC 28 September 2016, valid at six days lead time from the forecast initialized at 00:00 UTC 22 September 2016, as well as the ensemble mean absolute forecast error of the same date. Contour lines of geopotential height show two key features: A large gradient over the North Atlantic indicates the jet stream, a low-gradient region over eastern Europe indicates the remnants of a high-pressure system over Scandinavia that dominated northern European weather in the days before. The decay of this structure (referred to as "block") enabled the jet over the Atlantic to extend over Scandinavia, which steered the cyclone that developed in the wake of Karl into Norway, causing extreme precipitation.

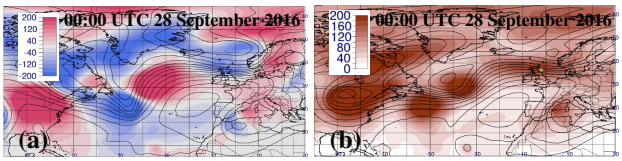

Fig. 9: **(a)** Geopotential height error (m) of control forecast, and **(b)** geopotential height mean absolute error (m), at 300 hPa with a lead time of 144h. Contours show geopotential height of control run **(a)** and analysis **(b)**.

Of interest are forecast errors over northern Europe associated with the decay of the block and the jet impinging on Norway. The error field exhibits very different structure in different members of the ensemble (not shown), yet still in the ensemble mean of the absolute errors, errors associated with this jet are very large, with a maximum over Scotland (Fig. 9b). It is a natural choice to first consider this error maximum; we hence aim at investigating the sensitivity of errors in this region to geopotential at earlier forecast times.

The user selects a seed location at 300 hPa over Scotland, at 00:00 UTC 28 September 2016, and the system computes an extended correlation clique with low stochastic variation (see Fig. 10) It captures the error region over Scotland and extends vertically throughout the entire atmosphere but does not include the southern Scandinavian region. The ESA signal (sensitivities) at the selected time (Fig. 10b) show a distinct dipole correlation pattern indicating that members with smaller forecast errors have lower (higher) geopotential height north (south) of the large gradient. Furthermore, the dipole pattern extends to the east of the large gradient. This pattern indicates that the actual jet was stronger and extended farther to the east than in the ensemble mean. The sensitivities, however, quickly vanish as we trace them backwards in time (illustrated in Fig. 10c for the positive ESA signal north of the large gradient). Evidently, the error maximum within the largest gradient is thus not associated with geopotential features at previous times earlier than 12-24h and no meaningful insight is gained into which earlier processes may have caused the error. As the region of large errors is located in a region of a strong gradient, however, the

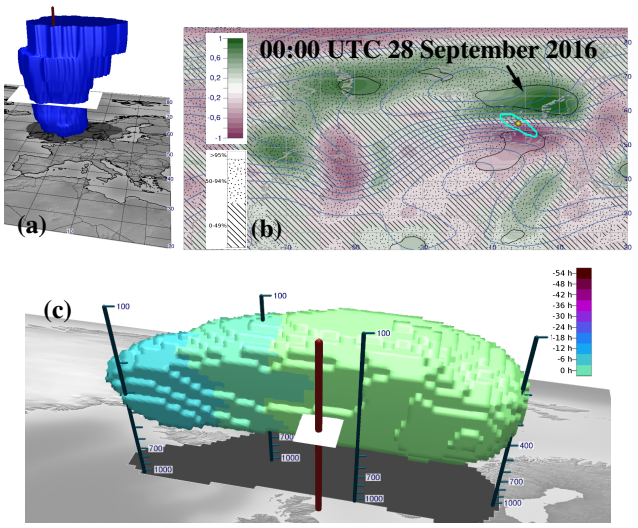

Fig. 10: (**a**) Correlation clique of geopotential height error, seeded at 300hPa over Scotland (pole and white cut). (**b**) ESA at 300hPa, using the clique from (a). (**c**) 3D swipe-path of sensitive region north of Great Britain (black arrow in (b)). The sensitivity signal vanishes after 18 h.

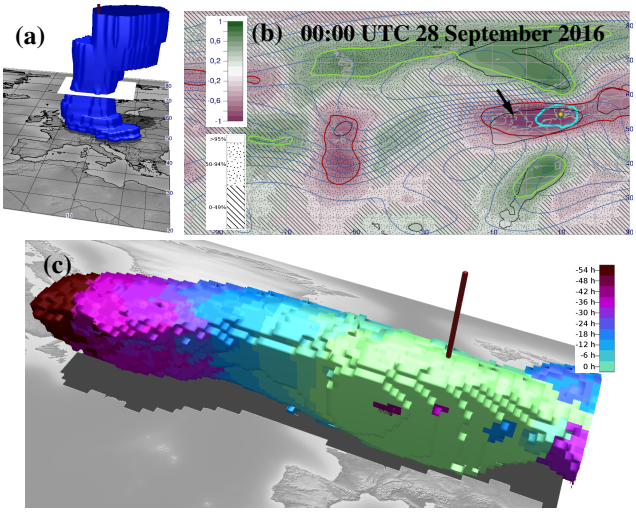

Fig. 11: **(a)** As Fig. 10a, but seeded over Denmark (pole). **(b)** ESA between geopotential height error and geopotential height at 300hPa using root mean square error of clique (turquoise) from **(a)** as input metric. Blue contours show mean pressure of geopotential height, stippling covers statistically insignificant regions to black contours marking 95% confidence. The red region indicated by a black arrow was later tracked. **(c)** Swipe-path of sensitivity feature over Denmark to correlation clique (a) seeded at dark red pole. Color changes in 6 hourly steps. Darker color indicates earlier valid times. Blue colors in the east show that the initial feature must have split before the second time step.

considered error maximum may be dominated by uncertainties in the north-south location of the jet.

Our interest here, however, is rather on the zonal extent of the jet and thus to investigate the situation further, we consider the lower-gradient region at the eastern end of the jet. A correlation clique in this region, centered at 300 hPa over Denmark (Fig. 11a), also yields a very coherent spatial structure, both in the horizontal and in the vertical, similar in this respect to the one associated with the error maximum discussed above. The ESA signal now exhibits a "tripole" structure with an elongated region of strong negative correlations overlapping the clique and positive correlations to the north and south. Hence, members with smaller forecast errors exhibit a stronger and more northerly jet and a stronger block to its east than the ensemble mean, and in addition more pronounced wave breaking over the western Mediterranean.

Fig. 11c shows the 3D swipe-path obtained from tracking the negative (red) ESA feature. The sensitivity pattern can be traced up- (westwards) and downstream (eastward) in time, respectively, with notably faster speed "upstream" than "downstream". This error pattern in the large-scale flow therefore has both an upstream component, as well as a more local "block" component. Importantly, the sensitivity pattern gets gradually more confined to the upper troposphere when tracing back in time, which emphasizes the importance of the jet structure over the North Atlantic two days before the development over Scandinavia. Focusing on this upstream propagation of the ESA feature, we find that the statistically-significant signal is lost at 18:00 UTC 25 September 2016 in the ridge over the western North Atlantic. This is the time at which the extratropical transition of Karl started to impact the ridge development ([37]). Inspecting the ESA map at this time (at 300 hPa, Fig. 12a), we note a statistically-significant positive signal near the base of the trough upstream of the ridge. Going further back in time reveals that this signal highlights persistently the region between the tropical cyclone Karl and the upstream trough in the preceding 24 h (illustrated at 12:00 UTC and 00:00 UTC 25 September 2016, Fig. 12b,c). This signal is consistent with the high sensitivity of the outcome of the extratropical transition to the occurrence of the tropical cyclone and the upstream trough [8].

Our approach facilitates the intuitive building of hypothesis chains. As one example, it is now of interest if the error identified in the ridge is indeed associated with tropical cyclone Karl during its extratropical transition. To this end, we select a clique of geopotential errors from the center of the negative ESA signal in the ridge at 00:00 UTC 26 September 2016 and investigate its correlation with low-level features (using 925hPa geopotential). At this time, the ridge error is associated with a statistically significant dipole in 925hPa geopotential elongated in the direction of Karl's track (not shown), indicating that ridge errors correlate with the position of Karl. This dipole pattern can be traced back for 96h (Fig. 13), using the positive part of the signal, until the start of the forecast. The signal, however, is statistically significant only in the first and last 24h of the considered time period. Still, Fig. 13b indicates that errors in the track of Karl lead to errors in the ridge formation over the North Atlantic, which in turn leads to errors in the large-scale flow over Scandinavia two days later. Application of our method thus greatly facilitates building a hypothesis chain that ultimately relates the extreme precipitation event in southern Norway to the evolution of tropical cyclone Karl. It is a promising task for future work to further elaborate on this hypothesis.

5.2 Smaller-scale perspective: Moisture flux

The second perspective focuses more directly on the extreme precipitation event, aiming at identifying the sensitivity of the forecast precipitation amount to uncertainties in the forecast moisture flux (the product of humidity and wind) at earlier times. When selecting a square target region enclosing a significant weather event, as in common meteorological workflows, a rather weak ESA signal is computed (see Fig. 4b,c). By using our workflow, the user has picked different seed locations (see Fig. 3), until a coherent correlation clique (Fig. 1a) yields clear, significant ESA signals (Fig. 4d). The interpretation of the signal is rather straightforward: The positive signal over southern Norway and upstream (to the west) reveals that stronger precipitation is associated with stronger antecedent moisture flux. Interestingly, our case does not exhibit a dipole pattern close to the region of extreme precipitation, which indicates sensitivity to the location of antecedent moisture flux and has often been found in other studies. Here, instead, the sensitivity is to the magnitude of the moisture flux over a relatively broad area.

The automated tracking clearly traces the signal back to the beginning of Karl's extratropical transition approximately 4 days earlier (Fig. 1c). The split-merge graph (Fig. 8b) and the mean tracking vectors in Fig. 1c indicate that the signal undergoes several merge and splitting

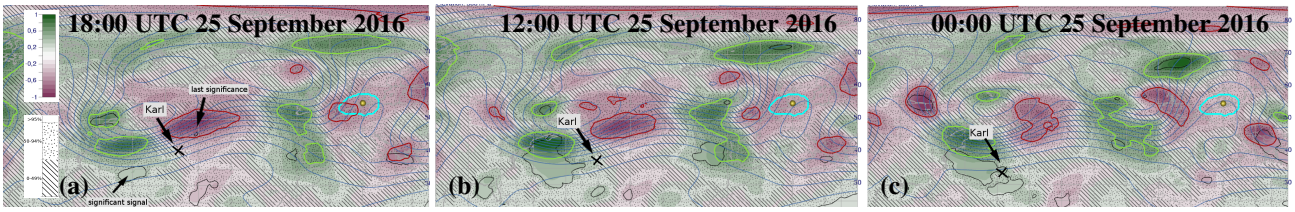

Fig. 12: ESA to clique over Denmark (cf. Fig. 11). **(a)** The last significant part of the tracked red region vanishes. A new significant signal is detected close to Karl. **(b,c)** It appears in earlier time steps as well, indicating a link between the error over Denmark and the early stage of Karl.

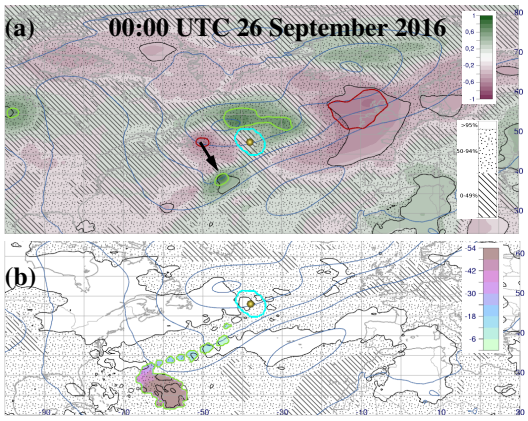

Fig. 13: **(a)** Sensitivities between geopotential height error at 300hPa (clique marked in turquoise) and geopotential height at 925 hPa. **(b)** Swipe path for the positive (green) sensitivity feature marked by the arrow in (a).

events during this time. In particular before 00:00 UTC 26 September 2016, the sensitivity regions are of relatively small scale before they merge into a spatially more coherent pattern (Figs. 8b, 1c). By inspecting the individual ESA maps we visually verified the automated tracking. The ESA signal found in the moisture flux thus corroborates the hypothesis chain developed using the geopotential correlations, i.e., that the extreme precipitation event in Norway is sensitively linked to the evolution of tropical cyclone Karl.

Finally, it is important to note that the statistically coherent structures in the moisture flux field are relatively small compared to the ones in the geopotential height field (Fig. 5), i.e., the moisture flux region exhibits more nontrivial smaller-scale structures. Hence, more atmospheric observations are required to adequately sample the sensitive moisture flux region. If targeted observations were collected to reduce the uncertainty in the forecast of the extreme precipitation event, the analysis suggest that it might be more efficient to sample the sensitive geopotential height region using less observations. Depending on the forecast horizon, the observations should be collected in the region between Karl and the upstream trough, to improve the forecast of Karl's track, or directly within the ridge after the completion of the ET.

5.3 User discussion

We discuss our first experiences with the method from the point of view of the meteorological domain experts in the author team. To our knowledge, this is the first interactive end-to-end workflow for ESA (providing guidance from selecting the region for J to tracking of sensitivities, including guidance on robustness of results and possibly further locations of interest, i.e., supporting hypothesis chain building). Compared to our current script-based workflow, ESA is therefore greatly facilitated during all necessary steps of the analysis. Definition of a suitable region for J using correlation cliques greatly facilitates selection of a meaningful (in a statistical sense) region. In our script-based workflow, region selection was guided by meteorological intuition. Now, the selection of the initial point is guided by intuition, the region is proposed by the automated method. The user can check if the proposed region contains the event of interest.

Tracking of sensitivity features and subsequent depiction of a swipe-path provides a quick, succinct overview of the evolution of the sensitivities, thus enabling fast judgement if further investigation of the sensitivities should be conducted. If yes, attention of the investigation is directed to the "end" (in a backward tracking sense) of the swipe-path, which can be the starting point of any further sensitivity analysis (in the context of building hypothesis chains). In particular, the 3D depiction of the swipe-path provides unprecedented insight into the evolution of the 3D structure of the sensitivities. The split-merge diagram provides guidance for the robustness of the signal. If many split and merge events are present, the user needs to evaluate the sensitivities carefully to judge their physical relevance; few, distinct events, on the other hand, may indicate physically meaningful evolutions. For example, the lack of split events between 28 and 26 September in the orange curve in Fig. 8b is guidance to interpret the track as physically meaningful backward propagation of the sensitivity signal from the initially selected weather event of interest to the Western Atlantic near Karl.

6 Conclusion

We have proposed a novel visual analysis workflow to facilitate an interactive analysis of sensitivities of a forecast metric J on another forecast field. Our workflow enables the user to interactively identify regions of intercorrelated grid points from which J is computed, and to automatically track features of high sensitivity through time. A "swipe-path" visualization showing the track of a sensitivity feature in time has been proposed that allows the user to immediately see geographical regions from which sensitivities originated. Swipe-paths are generated from user-selected features and can be displayed both in 2D and 3D. In particular, the novel interactive sensitivity tracking in 3D opens the door to analyses considering all spatial dimensions, as not possible with existing 2D meteorological workflows.

The workflow has been integrated into the open-source software Met.3D. Its benefit has been demonstrated with a real-world case study taken from ongoing analyses of the NAWDEX atmospheric field campaign. Compared to script-based tools commonly applied in the meteorological community, the workflow proposed here greatly simplifies ensemble sensitivity analysis by providing an interactive end-to-end workflow that encapsulates all steps required for sensitivity analysis in a single framework. At the same time, important information about the reliability of the results is provided, and a fully 3D analysis is facilitated. In the near future, the method will actively be used in further data analysis activities related to the NAWDEX campaign. Future work might include detailed statistics of sensitivity structures, a manual correction tool for the tracking and further enhancement of the swipe-path using glyphs to compensate for occlusion.

Acknowledgments

The research leading to these results has been done within the subprojects B5 and A4 of the Transregional Collaborative Research Center SFB/TRR 165 Waves to Weather funded by the German Research Foundation (DFG). Access to ECMWF prediction data has been kindly provided in the context of the ECMWF special project Support Tool for HALO Missions. We thank all the reviewers for their constructive criticism and valuable comments.

REFERENCES

[1] B. Ancell and G. J. Hakim. Comparing adjoint- and Ensemble-Sensitivity analysis with applications to observation targeting. *Monthly Weather Review*, 135(12):4117–4134, Dec. 2007. doi: 10.1175/2007mwr1904.1

[2] A. Anil Valsangkar, J. Merwin Monteiro, V. Narayanan, I. Hotz, and V. Natarajan. An exploratory framework for cyclone identification and tracking. *IEEE Transactions on Visualization and Computer Graphics*, PP:1–1, 02 2018. doi: 10.1109/tvcg.2018.2810068

[3] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004. doi: 10.1023/B:MACH.0000033116.57574.95

[4] G. Bradski and A. Kaehler. Opencv. *Dr. Dobb's journal of software tools*, 3, 2000.

[5] P. T. Bremer, G. Weber, J. Tierny, V. Pascucci, M. Day, and J. Bell. Interactive exploration and analysis of large-scale simulations using topology-based data segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 17(9):1307–1324, Sept 2011. doi: 10.1109/TVCG.2010.253

[6] C.-K. Chen, C. Wang, K.-L. Ma, and A. T. Wittenberg. Static correlation visualization for large time-varying volume data. In *Visualization Symposium (PacificVis), 2011 IEEE Pacific*, pp. 27–34. IEEE, 2011.

[7] H. Doraiswamy, V. Natarajan, and R. S. Nanjundiah. An exploration framework to identify and track movement of cloud systems. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2896–2905, Dec 2013. doi: 10.1109/TVCG.2013.131

[8] C. Evans, K. M. Wood, S. D. Aberson, H. M. Archambault, S. M. Milrad, L. F. Bosart, K. L. Corbosiero, C. A. Davis, J. R. Dias Pinto, J. Doyle, et al. The extratropical transition of tropical cyclones. Part I: Cyclone evolution and direct impacts. *Monthly Weather Review*, 145(11):4317–4344, 2017. doi: 10.1175/mwr-d-17-0027.1

[9] Z. Fang, T. Möller, G. Hamarneh, and A. Celler. Visualization and Exploration of Time-varying Medical Image Data Sets. In *Proceedings of Graphics Interface 2007*, pp. 281–288, 2007. doi: 10.1145/1268517.1268563

[10] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pp. 363–370. Springer, 2003. doi: 10.1007/3-540-45103-x_50

[11] T. Fiolleau and R. Roca. An algorithm for the detection and tracking of tropical mesoscale convective systems using infrared images from geostationary satellite. *IEEE Transactions on Geoscience and Remote Sensing*, 51(7):4302–4315, July 2013. doi: 10.1109/TGRS.2012.2227762

[12] A. Fofonov, V. Molchanov, and L. Linsen. Visual analysis of multi-run spatio-temporal simulations using isocontour similarity for projected views. *IEEE Transactions on Visualization and Computer Graphics*, 22(8):2037–2050, Aug 2016. doi: 10.1109/TVCG.2015.2498554

[13] A. V. Gambheer and G. S. Bhat. Life cycle characteristics of deep cloud systems over the Indian region using INSAT-1B pixel data. *Monthly Weather Review*, 128(12):4071–4083, 2000. doi: 10.1175/1520-0493(2000)129<4071:LCCODC>2.0.CO;2

[14] J. Hanley and R. Caballero. Objective identification and tracking of multicentre cyclones in the ERA-Interim reanalysis dataset. *Quarterly Journal of the Royal Meteorological Society*, 138(664):612–625, 2012. doi: 10.1002/qj.948

[15] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185 – 203, 1981. doi: 10.1016/0004-3702(81)90024-2

[16] D. Jen, P. Parente, J. Robbins, C. Weigle, R. Taylor II, A. Burette, and R. Weinberg. Imagesurfer: A tool for visualizing correlations between two volume scalar fields. In *Proceedings of IEEE Visualization*, pp. 529–536, 2004. doi: 10.1109/VISUAL.2004.46

[17] G. Ji and H.-W. Shen. Feature tracking using earth mover's distance and global optimization. In *Pacific Graphics*, vol. 2, 2006.

[18] M. Kern, T. Hewson, F. Sadlo, R. Westermann, and M. Rautenhaus. Robust detection and visualization of jet-stream core lines in atmospheric flow. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):893–902, 2018. doi: 10.1109/tvcg.2017.2743989

[19] A. Kumpf, B. Tost, M. Baumgart, M. Riemer, R. Westermann, and M. Rautenhaus. Visualizing confidence in cluster-based ensemble weather forecast analyses. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):109–119, 2018. doi: 10.1109/tvcg.2017.2745178

[20] M. Leutbecher and T. N. Palmer. Ensemble forecasting. *Journal of Computational Physics*, 227(7):3515–3539, 2008. doi: 10.1016/j.jcp.2007.02.014

[21] T. Liebmann, G. H. Weber, and G. Scheuermann. Hierarchical Correlation Clustering in Multiple 2D Scalar Fields. *Computer Graphics Forum*, 2018. doi: 10.1111/cgf.13396

[22] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, vol. 81, pp. 674–679, 1981.

[23] L. Magnusson. Diagnostic methods for understanding the origin of forecast errors. *Quarterly Journal of the Royal Meteorological Society*, 143(706):2129–2142, July 2017. doi: 10.1002/qj.3072

[24] S. J. Majumdar. A review of targeted observations. *Bulletin of the American Meteorological Society*, 97(12):2287–2303, Apr. 2016. doi: 10.1175/bams-d-14-00259.1

[25] Met.3D contributors. Met.3D. `http://met3d.wavestoweather.de`, 2018. Accessed 31 March 2018.

[26] C. Muelder and K. L. Ma. Interactive feature extraction and tracking by utilizing region coherency. In *2009 IEEE Pacific Visualization Symposium*, pp. 17–24, April 2009. doi: 10.1109/PACIFICVIS.2009.4906833

[27] T. Pfaffelmoser and R. Westermann. Visualization of Global Correlation Structures in Uncertain 2D Scalar Fields. *Computer Graphics Forum*, 31:1025–1034, 2012. doi: 10.1111/j.1467-8659.2012.03095.x

[28] T. Pfaffelmoser and R. Westermann. Correlation visualization for structural uncertainty analysis. *International Journal for Uncertainty Quantification*, 3:171–186, 2013. doi: 10.1615/int.j.uncertaintyquantification.201200393

[29] M. Rautenhaus, M. Böttinger, S. Siemen, R. Hoffman, R. M. Kirby, M. Mirzargar, N. R'ober, and R. Westermann. Visualization in meteorology - a survey of techniques and tools for data analysis tasks. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2017. doi: 10.1109/TVCG.2017.2779501

[30] M. Rautenhaus, M. Kern, A. Schäfler, and R. Westermann. Three-dimensional visualization of ensemble weather forecasts – Part 1: The visualization tool Met.3D (version 1.0). *Geosci. Model Dev.*, 8(7):2329–2353, 2015.

[31] F. Reinders, F. H. Post, and H. J. Spoelder. Visualization of time-dependent data with feature tracking and event detection. *The Visual Computer*, 17(1):55–71, Feb 2001. doi: 10.1007/PL00013399

[32] D. P. Retchless and C. A. Brewer. Guidance for representing uncertainty on global temperature change maps. *International Journal of Climatology*, 36(3):1143–1159, 2016. doi: 10.1002/joc.4408

[33] H. Saikia, H.-P. Seidel, and T. Weinkauf. Fast similarity search in scalar fields using merging histograms. In H. Carr, C. Garth, and T. Weinkauf, eds., *Topological Methods in Data Analysis and Visualization*, pp. 121–134. Springer International Publishing, Cham, 2015. doi: 10.1007/978-3-319-44684-4_7

[34] H. Saikia and T. Weinkauf. Global feature tracking and similarity estimation in time-dependent scalar fields. *Computer Graphics Forum*, 36(3):1–11, 2017. doi: 10.1111/cgf.13163

[35] R. Samtaney, D. Silver, N. Zabusky, and J. Cao. Visualizing features and tracking their evolution. *Computer*, 27(7):20–27, July 1994. doi: 10.1109/2.299407

[36] N. Sauber, H. Theisel, and H. Seidel. Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Transactions on Visualization and Computer Graphics*, 12:917–924, 2006. doi: 10.1109/tvcg.2006.165

[37] A. Schäfler, G. Craig, H. Wernli, P. Arbogast, J. D. Doyle, R. McTaggart-Cowan, J. Methven, G. Rivière, F. Ament, M. Boettcher, M. Bramberger, Q. Cazenave, R. Cotton, S. Crewell, J. Delanoë, A. Dörnbrack, A. Ehrlich, F. Ewald, A. Fix, C. M. Grams, S. L. Gray, H. Grob, S. Gro, M. Hagen, B. Harvey, L. Hirsch, M. Jacob, T. Kölling, H. Konow, C. Lemmerz, O. Lux, L. Magnusson, B. Mayer, M. Mech, R. Moore, J. Pelon, J. Quinting, S. Rahm, M. Rapp, M. Rautenhaus, O. Reitebuch, C. A. Reynolds, H. Sodemann, T. Spengler, G. Vaughan, M. Wendisch, M. Wirth, B. Witschas, K. Wolf, and T. Zinner. The north atlantic waveguide and downstream impact experiment. *Bulletin of the American Meteorological Society*, 2018. doi: 10.1175/BAMS-D-17-0003.1

[38] D. Silver and X. Wang. Tracking and visualizing turbulent 3d features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):129–141, Apr 1997. doi: 10.1109/2945.597796

[39] B. S. Sohn and C. Bajaj. Time-varying contour topology. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):14–25, Jan 2006. doi: 10.1109/TVCG.2006.16

[40] J. Sukharev, C. Wang, K.-L. Ma, and A. Wittenberg. Correlation study of time-varying multivariate climate data sets. In *Proceedings of IEEE Pacific Graphics 2009*, pp. 161–168, 2009. doi: 10.1109/pacificvis.2009.4906852

[41] R. Swinbank, M. Kyouda, P. Buchanan, L. Froude, T. M. Hamill, T. D. Hewson, J. H. Keller, M. Matsueda, J. Methven, F. Pappenberger, M. Scheuerer, H. A. Titley, L. Wilson, and M. Yamaguchi. The TIGGE project and its achievements. *Bulletin of the American Meteorological Society*, 97(1):49–67, Mar. 2015. doi: 10.1175/bams-d-13-00191.1

[42] R. D. Torn and G. J. Hakim. Ensemble-based sensitivity analysis. *Monthly Weather Review*, 136(2):663–677, 2008. doi: 10.1175/2007mwr2132.1

[43] R. D. Torn, J. S. Whitaker, P. Pegion, T. M. Hamill, and G. J. Hakim. Diagnosis of the source of GFS medium-range track errors in Hurricane Sandy (2012). *Monthly Weather Review*, 143(1):132–152, 2015. doi: 10.1175/MWR-D-14-00086.1

[44] H. Wernli and C. Schwierz. Surface cyclones in the ERA-40 dataset (1958-2001). Part I: Novel identification method and global climatology. *Journal of the Atmospheric Sciences*, 63(10):2486–2507, 2006. doi: 10.1175/JAS3766.1

[45] W. Widanagamaachchi, C. Christensen, V. Pascucci, and P. T. Bremer. Interactive exploration of large-scale time-varying data using dynamic tracking graphs. In *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 9–17, Oct 2012. doi: 10.1109/LDAV.2012.6378962

[46] H. Zhang, Y. Hou, D. Qu, and Q. Liu. Correlation visualization of time-varying patterns for multi-variable data. *IEEE Access*, 4:4669–4677, 2016. doi: 10.1109/ACCESS.2016.2601339

[47] M. Zheng, E. K. M. Chang, and B. A. Colle. Ensemble sensitivity tools for assessing extratropical cyclone intensity and track predictability. *Weather and Forecasting*, 28(5):1133–1156, May 2013. doi: 10.1175/waf-d-12-00132.1

Vision, Modeling, and Visualization (2019)
H.-J. Schulz, M. Teschner, and M. Wimmer (Eds.)

Cluster-based Analysis of Multi-Parameter Distributions in Cloud Simulation Ensembles

Alexander Kumpf, Josef Stumpfegger, Rüdiger Westermann

Technical University of Munich (TUM), Germany

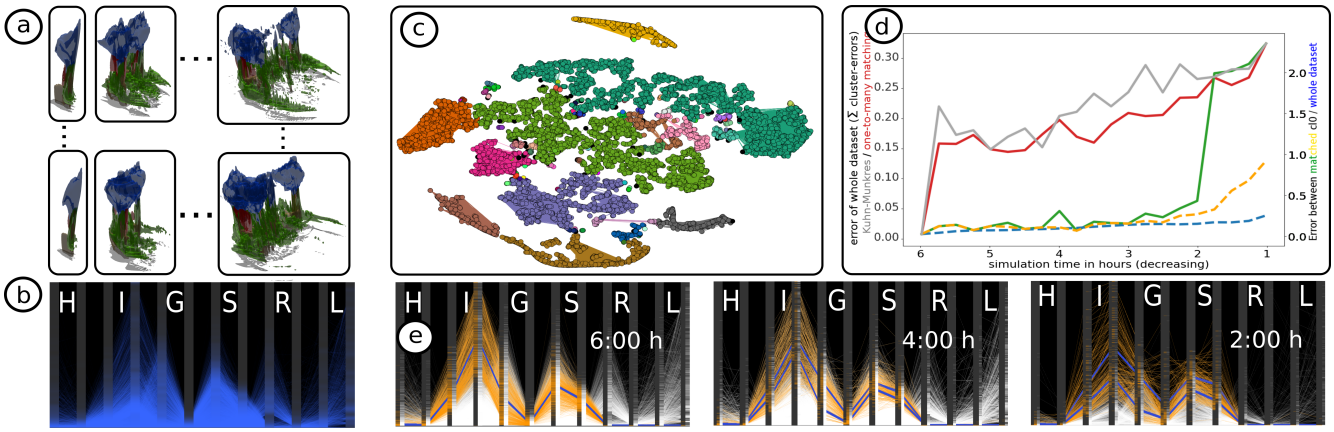

Figure 1: Visual analysis of a time-varying multi-parameter cloud ensemble. (a) Iso-surfaces of liquid water content (L, green), ice (I, blue), and graupel (G, red). (b) Parallel coordinates plot rendered on black background of per-voxel parameter values including attribute density histograms. (c) Voxel clusters determined via ensemble clustering using multiple k-Means clustered t-SNE projections. Voxels in the same cluster but separated in the t-SNE embedding are connected via lines. (d) Distribution-based matching of clusters over lead time, starting from 6 hours forecast. Colored curves show the matching error for different matching strategies and error metrics. (e) Selected cluster at 6 hours lead time (orange) and clusters matched to it at 4 and 2 hours lead time are highlighted in the parallel coordinates plot.

Abstract

The proposed approach enables a comparative visual exploration of multi-parameter distributions in time-varying 3D ensemble simulations. To investigate whether dominant trends in such distributions occur, we consider the simulation elements in each dataset—per ensemble member and time step—as elements in the multi-dimensional parameter space, and use t-SNE to project these elements into 2D space. To find groups of elements with similar parameter values in each time step, the resulting projections are clustered via k-Means. Since elements with similar data values can be disconnected in one single projection, we compute an ensemble of projections using multiple t-SNE runs and use evidence accumulation to determine sets of elements that are stably clustered together. We build upon per-cluster multi-parameter distribution functions to quantify cluster similarity, and merge clusters in different ensemble members. By applying the proposed approach to a time-varying ensemble, the temporal development of clusters, and in particular their stability over time can be analyzed. We apply this approach to analyze a time-varying ensemble of 3D cloud simulations. The visualizations via t-SNE, parallel coordinate plots and scatter plot matrices show dependencies between the simulation results and the simulation parameters used to generate the ensemble, and they provide insight into the temporal ensemble variability regarding the major trends in the multi-parameter distributions.

1. Introduction

Ensemble weather forecasting is well established in meteorology to estimate the uncertainty that is present in numerical weather predictions. Ensemble methods perform multiple simulations using perturbed initial conditions or different forecast models, to predict possible future states of the atmosphere. Analysis of the temporal evolution and variability of an ensemble forecast is then used to estimate the likelihood of certain weather events.

Ensemble methods are also used to analyze the effect of simulation parameters on the simulated weather events, by systematically

perturbing these parameters and running the simulation again using the perturbed configurations. The analysis, then, requires more than detecting similarities and differences between pairs of simulation results. Beyond that, coherent predictions, localized or over the entire domain, across many results need to be determined and put into relation to the input parameter values that have caused these situations. For an improved assessment of these predictions, they need to be quantified and visualized in context to each other.

We propose a visual analysis approach to help addressing this task, and use this approach to analyze an ensemble of time-varying 3D cloud simulations [WBJ*18]. Each simulation is carried out on a regular voxel grid, and for each voxel a 12D parameter vector containing quantities like water, ice, graupel, and hail content is simulated. As we seek to compare clouds of different size, shape and position, location-based approaches, i.e., computing statistical measures over the values at a single voxel, are not useful. For instance, if a cloud does not change with respect to its physical composition but simply moves in space, similarity measures invariant under such transformations need to be used.

Parallel coordinate (PC) plots can, in principle, be used for this purpose, by drawing one line strip for each voxel. However, it is difficult to reveal lower-dimensional manifold structures via PC plots, and the visualization becomes quickly cluttered when many elements are drawn simultaneously. Clustering, on the other hand, can determine sub-groups of elements with similar parameter values, providing a condensed data representation that facilitates a distribution-based analysis. Clustering in high-dimensional (HD) parameter space, however, becomes difficult due to the inherent sparsity of the data space and the difficulty to select an appropriate clustering algorithm and its parametrization.

Dimensionality reduction techniques can be used to address these problems. For instance, t-Distributed Stochastic Neighbor Embedding (t-SNE) [MH08] tries to preserve locality by placing similar elements close to each other in a low-dimensional subspace. Since lower-dimensional manifold structures in the original data are preserved, especially density-based clustering algorithms like DBSCAN [EKS*96], which focus on "reachability" rather than distance, show very good results if the right parametrization is used. However, since the parametrization needs to be adapted for every projection, in our current scenario the application of DBSCAN is not feasible.

1.1. Contribution and method overview

The proposed approach detects stable clusters of data points in a HD parameter space. It uses this information to enable a cluster-based analysis of the variability of ensembles of multi-parameter simulations, and to reveal dependencies of the simulation results on the initial simulation parameters. An overview of this approach is given in Fig. 2. By variations of a set of input parameters τ_i, an ensemble of multi-parameter simulations is computed. Simulation elements are interpreted as data points in the multi-dimensional parameter space, and they can be visualized using standard visualization techniques like volume rendering and PC plots.

Then, dimensionality reduction via t-SNE projects the data points into 2D. In this way, many of the local neighborhoods in the data are preserved and sub-manifolds in HD space become connected structures in 2D space. To avoid the shortcomings of DBSCAN in the current scenario, the projected points are clustered using k-Means, and the resulting clusters are put into relation using their variability over the ensemble.

Dimensionality reduction techniques like t-SNE, however, sometimes need to split a connected subgroup to compute the 2D embedding. Where these splits occur depends on the specific parametrization of the used technique. For instance, t-SNE is often used with random initial locations of projected objects as its seed configuration which are considered by gradient descent optimization. Therefore, when t-SNE is run with different input parameterizations, connected subgroups can be split in many different ways.

On the other hand, similar points should be placed close to each other most of the time over all projections, regardless of the specific initial parametrization. Thus, we compute many projections using different parameterizations and merge the clusterings which are obtained via k-Means into one final clustering. To visualize stable subgroups, the projection representing best the final clustering is picked, and cluster membership information per data point is encoded via colors. Additionally, some of the points are connected via lines to indicate where clusters were cut in the selected projection but can be assumed connected over all projections. To further analyze the distribution of parameter values in a cluster, they are displayed via PC plots, augmented by per-parameter distributions and overlayed representatives for selected clusters. Per-cluster distributions are then represented via cumulative distribution functions (CDF), and the differences in their integrals are used as similarity measure. This enables to match different clusters and find similarities across time steps and ensemble members. The similarity between ensemble members is put in context with the initial simulation parameters via scatter plot matrices.

In particular, the following contributions are made:

- A method to determine stable clusters in multi-parameter data sets, using t-SNE and k-Means-based ensemble clustering.
- A distribution-based similarity metric for clusters of multi-parameter data points.
- The application of cluster-based analysis of multi-parameter distributions to a time-varying multi-parameter 3D cloud ensemble, hinting on the effect of simulation parameters on weather forecast variability.

On a technical side, we provide a highly efficient GPU implementation of PC plots with embedded line and density histograms capable of plotting millions of multi-parameter data elements per second, including instant color variations to highlight selected clusters. In the 2D t-SNE view, multiple interaction possibilities are available to select and display single clusters, similar data elements, etc., over different projections, time steps, and ensemble members.

2. Related work

In our scenario, each ensemble member is comprised of a set of simulation elements with multiple parameter values. These HD data points are projected into 2D using t-SNE [MH08]. Some recent surveys [KH13; LMW*16] give thorough overviews of visualization techniques for multi-parameter data. In combination with

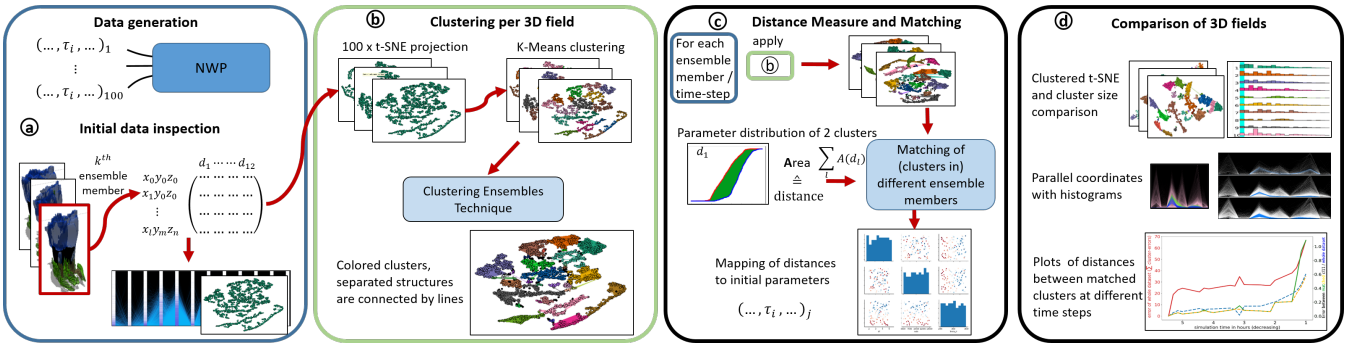

Figure 2: Proposed workflow for analyzing an ensemble of 3D multi-parameter simulations, each simulation parameterized by input parameters τ_i. **(a)** Upon using standard visualization techniques like volume rendering and PC plots, **(b)** k-Means clusterings on multiple t-SNE projections are combined to obtain a stable clustering of data points in a single 3D data set. The best embedding is determined for visualization. **(c)** For all ensemble members and time steps, clusters are matched based on their CDFs. A scatter plot matrix indicates relations between input parameters and multi-parameter distributions. **(d)** Visualization of the (temporal) variability of simulations using clustered t-SNE, cluster bar charts, PC plots, histograms, and line charts.

dimensionality reduction, clustering is often used to identify groups of points lying close together in the low-dimensional space or forming coherent structures in this space. Wenskovitch et al. [WCR*18] discuss the combination of dimensionality reduction and clustering techniques and provide recommendations for their concurrent use.

Since our technique analyzes an ensemble of 2D point sets (after multi-parameter simulation elements have been projected into 2D), it is related to ensemble visualization techniques. Most works in ensemble visualization address ensembles of physical fields, or features derived from such fields, with the focus on the extraction and visual encoding of their variability. To the best of our knowledge, visual analysis techniques for ensembles of 2D points are not existing, yet a number of techniques have addressed aspects related to ensembles that are also relevant in our work. Parametric statistical distributions and distribution shape descriptors for scalar-valued ensembles were presented by Love et al. [LPK05]. Different variants of confidence regions were introduced to represent the major geometric trends in ensembles of iso-contours and streamlines [WMK13; MWK14; FBW16; FKRW16]. Demir et al. [DJW16] proposed a closest-point representation to convey the central tendency of an ensemble of multi-dimensional shapes. In a number of works, scalar- and vector-valued ensemble fields were modeled via mixtures of probability density functions to compactly classify complex distributions and their evolution over time [LLBP12; JDKW15; DS15; WLW*17]. Demir et al. [DDW14] visualize distributions of linearized 3D data points with bar-charts. Hummel et al. [HOGJ13] analyze the spread of particle trajectories in an ensemble of vector fields to reveal the transport variability. Poethkow and Hege [PH13] and Athawale et al. [ASE15] use location-wise estimators of non-parametric distributions from ensemble members to estimate the spread of surface and vector field features. Recently, Hazarika et al. [HBS17; HDSC19] presented a copula-based framework for large multivariate datasets, where they partition the domain and compute statistical quantities over those parts.

Alternatively, clustering has been used to group ensemble members regarding similar data characteristic [BM10; OLK*14; FBW16]. While these techniques compare ensemble members to each other, our approach aims at finding groups of elements in each member which remain "close" to each other in all members, and then match different clusterings to each other. Strehl and Ghosh [SG02] apply different clustering techniques to one single ensemble, and combine the results into a single clustering. Different clustering ensemble techniques, i.e., techniques that combine multiple clusterings of one data set into a single clustering, are discussed by Vega-Pons et al. [VR11]. From multiple k-Means clusterings, Fred and Jain [FJ05] generate a co-association matrix, containing the fraction of times two points were placed into the same cluster. Applying clustering on this matrix leads to the final result. Kumpf et al. [KTB*18] use multiple k-Means clusterings on ensemble data, where they vary the clustering domain to generate a clustering ensemble. Ferstl et al. [FKRW17] cluster different time steps of the same ensemble in a hierarchical way to convey the change of clusters over time. For the clustering of genomic data, Lex et al. [LSP*10] introduce extended PC plots to compare different clusterings and analyze the quality of cluster assignments.

Related to our approach are also techniques which aim to find projections that best represent the structures in HD data, by using quality measures for projections [FT74; HA85]. Even though the goal of these techniques is different to ours, as we do not attempt to find the best projection for a given dataset, proposed measures indicate the (dis-)similarity between projections and might be used for robustness analysis as well. Examples include vector distance measures for HD feature descriptors [BvLBS11] and feature vectors derived from point-wise distance matrices [JHB*17], as well as measures using matrix norms to quantify the dissimilarity of multivariate projections invariant to affine transformations [LT16].

3. Data

We apply our cluster-based approach to analyze the multi-parameter distributions in a numerical simulation of a growing thunderstorm cloud [WBJ*18]. The data set comprises an ensemble of 100 simulation runs of a single convective cloud in the 3D

atmosphere, simulated over a time span of 6 hours in time steps of 15 minutes. 4 members have been excluded due to corrupted values resulting in a total of 96 members. At each 3D position, 12 parameters—such as water content, ice-water content, number of water particles, number of ice particles—are given. The numerical simulation depends on 6 input parameters such as wind-sheer, which influence the outcome of the simulation.

Due to computational reasons, only every 4^{th} value in the two spatial coordinates x and y, and every 3^{rd} value in the vertical is used, resulting in roughly 10000 data points per time step. Values of each attribute are normalized to the range of $[0,1]$ excluding the ones with norm smaller 0.1 in order to shift the focus of the analysis away from almost empty voxels. No restrictive assumptions about the structure of clusters are made. Further, since quantities within clouds transition smoothly between states, as water slowly starts to freeze with decreasing temperature, elongated structures are expected at the least.

4. Dimensionality reduction and k-Means clustering

For dimensionality reduction, the method t-SNE (Fig. 4a) is used. Note that while variants such as Hierarchical Stochastic Neighbor Embedding [PHL*16] can be used as well, deterministic dimensionality reduction techniques like principal component analysis (PCA) are not suitable in the current scenario. Points can be misplaced due to variation in others than the principal components subspace used for projecting. Re-running PCA would not change that. Multi-dimensional scaling [KW78], on the other hand, seeks at preserving distances over the whole domain, thus making it difficult to maintain local structures in the generated 2D embeddings.

In a single projection, the distances between data points can be significantly distorted depending on the parametrization of the used projection technique. The reason is that dimensionality reduction techniques need to cut manifold structures in the HD space to embed them into 2D. For instance, when projecting a sphere there is no 2D embedding that can avoid placing non-neighboring points close together or flattening the sphere so that neighboring points become distant to each other. This problem can be addressed by running t-SNE many times with different parameters or random initialization, so that cuts are introduced at different locations and the neighborhood relations are maintained in most projections. Each projection can be clustered individually, and the clustering results can be further analyzed to extract sets of data points that are coherently assigned to the same cluster. In addition, however, the individual clusterings need to create consistent results for different ensemble members and time steps, to allow for a later comparison of these results. It is clear that this cannot be achieved by tweaking the parameters of each individual clustering. Due to this requirement, density-based clustering approaches (e.g., DBSCAN [EKS*96]) are not suitable in the current application. The clustering results of these algorithms are rather sensitive to variations in the distances between projected data points, which, as described before, can happen to a certain extent in different t-SNE projections. It is worth noting that the same problem occurs when clustering is applied to the original HD data point, as shown in Fig.3a.

The clustering algorithm k-Means, on the other hand, always

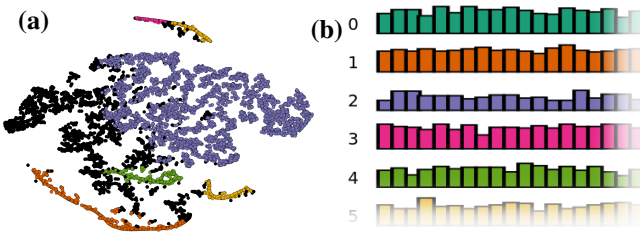

Figure 3: **(a)** High-dimensional DBSCAN clustering with parameters $\varepsilon = 0.04$ and $N = 30$ color coded on t-SNE projected points. The blue cluster and noise in black dominate the clustering. **(b)** Matched clusters of k-Means clustered t-SNE projections. Each column represents one clustering, the height of the bar encodes the number of points contained in a cluster.

generates a predefined number of clusters. Though, the clusters are convex and can hence put two independent elongated structures into the same cluster or cut them at an arbitrary location. However, in different projections these cuts are introduced at different locations; furthermore, if these structures are not adjacent in the original data, a different t-SNE projection is likely to place them far apart from each other in the computed 2D embedding. Therefore, only points that are neighbored in the HD space should be in the same cluster in most of the projections, thus overcoming the convexity requirement of k-Means clusters.

Due to the aforementioned issues, we use t-SNE, with default perplexity of 30, and k-Means, with $k = 16$ clusters, in our analysis. The perplexity parameter controls the size of the local neighborhood that should be preserved. In all of our experiments, the resulting projections looked reasonable, showing frequent yet spurious variations that support our envisioned consistency analysis. The number of clusters for k-Means has to be set in relation to the number of projections used. The higher the number of different t-SNE projections, the more clusters can be used to obtain more detailed results. The same parameters are used for all ensemble members and time steps in order to preserve comparability.

4.1. Combination of clusterings

The ensemble of clusterings that is generated by clustering multiple t-SNE projections separately is aggregated to obtain a final clustering. Points that are stably clustered together are extracted by using the so-called co-association matrix C [FJ05]. Each entry, C_{ij} counts how often point p_i and p_j are in the same cluster, finally normalized by dividing through the number of clusterings. For every point, the clique of points with high mutual similarity is searched in C. The similarity threshold is set to $\alpha = 0.9$, meaning that every pair of points in the same clique is clustered together in at least 90% of the single k-Means clusterings. For clique construction, we use algorithm 1 as proposed in Kumpf et al. [KRRW19]. In a final step, illustrated in Fig. 5, points are merged based on their cliques in a greedy-like manner using region growing. Starting with the point with the largest clique, recursively, all points therein and in their cliques are merged. Once no more points can be added, a cluster is formed and recursive merging is continued with the remaining points, starting again with the one with the largest clique.

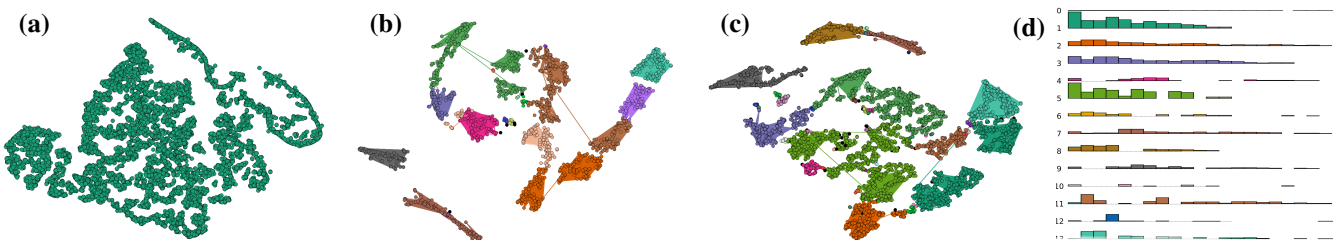

Figure 4: (a) 2D t-SNE projection of voxels. Connected structures motivate the use of clustering. (b-c) Final clustering resulting from the combination of 100 k-Means clustered t-SNE projections. Lines connect distant points in clusters, points of the same cluster have the same color, with noise in black. The projections with shortest line sums were chosen for the 2D embedding. Time steps **(b)** 2:00h **(c)** 4:00h are shown. **(d)** Each column represents a clustering of one time step starting with the latest, where height encodes the cluster size.

input: i_0, C, α **output:** L
$i = i_0$; $I = \{0,\dots,n\}$; $L = \{\}$;
while $(|L| < |I|)$ **do**
  $L = L \cup \{i\}$; $I = I \backslash \{i\}$;
  $I = \{j \in I | C_{ij} > \alpha\}$; $i = \arg\max_{j \in I} C_{ij}$;
end

Algorithm 1: Generation of point clique for p_{i_0} (similar to [KRRW19]) using the co-association matrix C using a pairwise similarity threshold α. Resulting cliques contain points with mutual similarity greater than α.

The merging algorithm depends on the order in which points are traversed. However, using random initial points has lead to similar results in all our experiments. To better understand the merging process, k-Means clusters can be matched as in [KRRW19] over all projections using the Kuhn-Munkres algorithm [Kuh55] (Fig. 3b). Single clusters can be selected and their position is displayed over different t-SNE projections. Fixing the projection and showing clusters from other projections is available as well, revealing neighboring points in other projections which were placed apart in the current one. Furthermore, one can search for points which were always assigned to a selected cluster or highlight points which were almost always together in the same cluster, which greatly helps understanding the effect of the merging threshold α. These interactions help understanding the quality and variance of the clustering ensemble and can be performed before comparing datasets. Later, it can be used to see the evolution of single clusters or understand why certain structures fall into different clusters.

The final obtained clusters represent points which lie in the same structure in the t-SNE plot and are therefore expected to form structures in HD parameter space. An example is given in Fig. 1c. Note that the number of clusters can now exceed 16. Points are colored according to their cluster ID, using black for noise. Additionally, lines between points, and in the color of these points, are drawn if the points are far apart. Short lines can be filtered out interactively.

Lines connecting adjacent points but located far away from each other in the current projection are generated as a byproduct during the merging step. Whenever points are merged, lines from the parent to all children are saved and used later in the final clustering. This facilitates the identification of clusters which were torn apart in the currently selected projection and attenuates the problem of finding a sufficiently large number of distinguishable cluster colors, i.e., clusters that are far apart from each other and not connected are different regardless their assigned color.

To visualize the cluster information, one projection has to be selected as a representative 2D embedding of the data points. We use a projection instead of other cluster visualizations, since these projections preserve the spatial relationships between points and clusters. Following the intuition that points in clusters should be located close to each other, we select the projection with the minimal sum of filtered line-lengths between the data-points. The final clustering result is then investigated further using PC plots, i.e., to compare different time steps and ensemble members.

4.2. Matching and comparing clusterings

A relation between final clusters of different ensemble members is established by comparing the distributions of their parameter values. For each parameter, a CDF is constructed [HDSC19]. To compare two clusters, for each parameter the area between their CDFs is computed (see Fig. 6) and summed up. Since this similarity measure depends only on the distribution of the parameter values, it can be used to compare clusters with vastly different size. Since this could become as extreme as matching two points to the biggest clusters, we penalize differences of a factor 10 and higher by adding a linear factor of

$$\text{penalty}_{cl_i,cl_j} = \max\left(0, \left(\frac{\max(|cl_i|,|cl_j|)}{\min(|cl_i|,|cl_j|)} - 10\right) \cdot 0.01\right),$$

to the cluster distance. Here, $|cl_i|$ denotes the size of cluster i. This similarity measure can be used to determine the similarity between two data sets, i.e., two ensemble members or different time steps of the same member, and to compare two clusters.

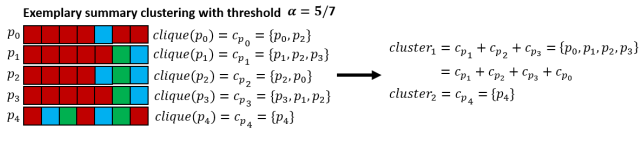

Figure 5: Clustering to combine the clustering ensemble. Color denotes the cluster ID for each point p_i for 7 clusterings. Note that clique(p_2) does not contain p_1 as $\text{sim}(p_0, p_1) < 5/7$. For this example only, the clique threshold was set to $\alpha = 5/7$.

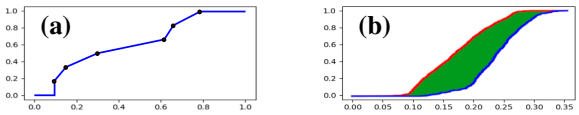

Figure 6: **(a)** Construction of a Cumulative Distribution Function (CDF) based on 6 sample points. **(b)** Comparison of the parameter distribution of 2 clusters (red and blue) by constructing a CDF. The area between the lines defines the distance between the clusters.

Clusters are then matched using either the Kuhn-Munkres-algorithm [Kuh55], resulting in a one-to-one matching between $\min(|cl_i|, |cl_j|)$ clusters, or alternatively, in a one-to-many matching, where each cluster of one clustering is matched to the best fitting cluster of the other clustering. For single clusters, the results match most of the time as can be seen by the green and yellow line in Fig. 1d. However, mismatches can lead to significantly higher differences, which is why we favor the one-to-many matching.

Summing over the $ln(|cl_i|)$-weighted differences between clusters and normalizing them by the summed weights serves as a second distance measure for data sets (Fig. 1d,red). When multiple *clusterings* are matched to one reference (Sec. 4.4), the sizes of the clusters in the reference clustering are used. Cluster sizes are not important to us. However, since the number of clusters should not affect the measure significantly, less weight is given to smaller clusters to prevent them from dominating the measure. Alternatively, only the x largest matched clusters can be used to compute the distance measure. In our experiments, both strategies are used, where in the latter clusters with less than 25 points are not matched.

4.3. Parallel Coordinates

Based on the matching errors, the sizes of clusters over time (see Fig. 4d), and by using PC plots, data sets can be compared to each other and similarities in parameter distributions can be investigated.

PC plots offer a direct visualization of HD data points. Our implementation uses the Vulkan graphics API, to enable the efficient visualization of huge numbers of multi-parameter data points. On our target architecture, an NVIDIA GTX 1070, up to 5 million 12D data points can be drawn per second. Basic functionality like blending and the reordering of axes can be used to get a first impression of the data. Histograms per displayed cluster on the coordinate axes ease the comparison of value distributions. Optionally, lines can be smoothed to better show densities. Mean and median lines can be drawn instead of whole clusters to avoid visual clutter.

4.4. Selection of reference

The presented analysis requires a reference dataset as starting point, to which others are compared to. Commonly, the simulation generated with best guessed initial parameters is used for that purpose, which is unfortunately not know for this dataset. Instead, the simulation generated with the median of all initial parameter configurations is investigated first. All initial parameter configurations are displayed in a scatter-plot-matrix (Fig. 8) were their distributions can be seen. The matching distances to the selected reference are displayed in color indicating which parameter changes lead to larger distances between the data sets. To gain an understanding

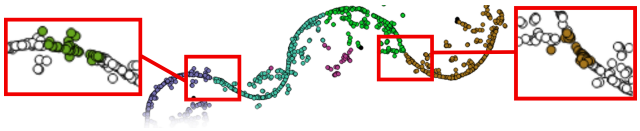

Figure 7: Reducing the merging threshold α to 0.8 creates connections between the clusters which leads to a merge.

for the data set, different members can be selected as reference to further investigate dependencies with initial parameters.

5. Use case

In the following, we describe the application of our approach to analyze the multi-parameter cloud ensemble described in Sec. 3. At the beginning, multiple time steps of a single ensemble member are visualized using iso-surfaces in single parameter fields (Fig. 1a). Despite the inherent occlusion effects, the overall structure of the clouds can already been observed: While wet quantities like liquid water (green, L) or rain (yellow, R) dominate in the lower altitudes, frozen quantities like ice (blue, I), graupel (red, G) and hail (brown, H) dominate in the upper atmosphere levels.

The shapes of the clouds change significantly over time, and they move over the domain, so that location-wise computation and comparison of data statistics is no option. Instead, we abstract from the 3D shape and perform the analysis using the distribution of parameter values as described. Firstly, PC plots are generated to obtain an initial estimate of the parameter distributions (Fig. 1b). By looping through the plots of all time steps of a selected ensemble member, the distribution variability over time is conveyed. The distributions seem to stay similar over all time steps, with the exception of strong hail (H), which is present only in later time steps. This is expected since ice-particles need some time to grow within the cloud. However, it is difficult to see whether the cloud forecasts are comprised of individual structures. To analyze this, the data is projected using t-SNE (Fig. 4a). The projections of simulation elements into 2D reveal many band structures and clusters of elements, yet it is impossible to conclude on which structures belong together and which are separated. After generating a stable clustering (explained in Sec. 4 and 4.1), clearly separated clusters appear (color coded in Fig. 1c). Connecting lines highlight where these structures where not cut in other projections, e.g., the rose cluster. Furthermore, small blue clusters of almost the same color can be differentiated, none of them connected via lines.

To investigate which clusters might merge due to a different merging threshold α, points can be picked interactively on the boundary between clusters and the effect of varying α can be seen (Fig. 7). When points from both sides pop-up, the clusters would merge. In this way, the cluster ensemble step and the degree of dissimilarity between clusters can be better understood. Further interaction mechanisms, e.g., selection and tracing of clusters, are provided as additional options to the user.

By using the proposed approach, points of specific clusters can be directly emphasized in the PC plot (Fig. 1e). For every quantity, there are two axes showing the weight and number of particles of that quantity in the corresponding simulation element. Elements in the orange cluster contain mostly ice (I), snow (S) and graupel (G).

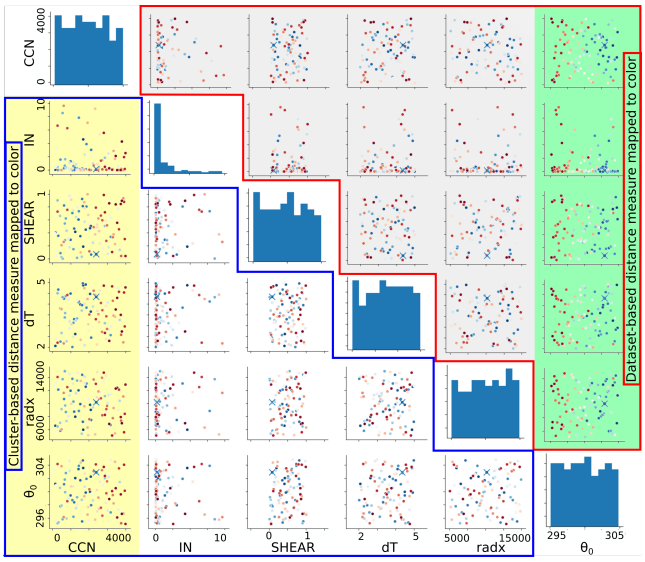

Figure 8: Matching distances to median simulation based on whole data set and on clusters, color coded on initial parameters, with low-to-high in blue-to-red. The data set- and cluster-based distance measures reveal correlations with θ_0 or CCN respectively.

Next, we cluster all time steps of the selected ensemble member and match clusters to the last by using the proposed many-to-one matching (see Sec.4.2). The matching errors (Fig. 1d) indicate that until time step 2:00h the distributions are very similar and the matching works well. For earlier time steps, the errors grow. Closer inspection of the worst matched clusters reveal that precipitation parameters are differently distributed in earlier time steps. When looping through the colored t-SNE plots—with matched clusters, similar structures can be observed over different time steps (Fig. 4). Caused by the many-to-one matching, some clusters become empty when no matching partner could be found (Fig. 4d). By focussing on the orange cluster to which two clusters were matched at time step 2:00h (Fig. 4b), its time evolution can be displayed using PC plots (Fig. 1e). The median or mean lines (blue lines) as well as a histogram bar per cluster and axis can be selected as well. It can be observed that the distribution of the cluster stays mainly the same, while the number of simulation elements decreases over time.

A similar analysis can be performed over all ensemble members. The proposed metrics can be used to find similar and dissimilar members. First results can be seen in Fig. 8, where matching distances are color coded on the initial parameters. A correlation between temperature (θ_0) and the distance between the whole data sets is directly visible and highlighted in green. When using the cluster based measure, correlation with cloud condensation nuclei (CCN), highlighted in yellow, becomes visible. This indicates that θ_0 changes the overall distributions while CCN changes the structures in the parameter space. Since all initial parameters were perturbed simultaneously, multi-dimensional dependencies have to be considered as well, which is left for future work.

Computation time: Performance is measured on an Intel® Xeon CPU 6 cores @3.5GHz. Preparing each data set takes around 35s, 70s for each t-SNE projection, and 0.5s for k-Means, all performed on one core. Combining the cluster ensemble takes 240s using all cores and cluster matching around 15s per dataset on one core. These pre-computations can be parallelized over the datasets.

6. Discussion and conclusion

Many steps of the proposed approach are dependent on parameters, albeit most of them are rather uncritical when kept constant over the whole analysis. Together, the number of k-Means clusters and projections define the granularity of the approach. More projections allow more k-Means clusters, the ratio of 100 projections to 16 clusters resolved the structures quite well in the presented data set. Multiple k-Means clusterings per projection could further reduce the number of projections needed.

The most critical parameter is the matching threshold α. Its impact can be seen in Fig. 7, where reducing α from 0.9 to 0.8 would merge the structure. A smaller threshold leads to bigger clusters. We advice to chose and fix this parameter once in the beginning after fixing all other parameters. That way, the merging stays consistent for all data sets. Further, the many-to-one matching corrects some undesired cluster splits. Matching successive time steps instead of to the last makes immediate changes visible. However, errors would propagate over time leading to a loss of overall context. Further, the approach relies on t-SNE's ability to project adjacent HD points close to each other most of the time. Sufficient variation in t-SNE projections and k-Means clustering is needed to extract structures of arbitrary shape. Alternatively to clustering voxels, one could cluster directly in the parameter space using subspace clustering methods. Optimally, those algorithms find clusters in all parameter-dimension combinations. Analyzing, comparing and matching those clusters would be a challenging task.

With the proposed method we were able to gain first insights into the parameter-value distributions of a time-dependent cloud ensemble data set. Cluster ensemble techniques on k-Means clustered t-SNE plots proved to be a valid way for extracting structures from that data set, which could be found in other time steps as well using a CDF based distance measure. While the clouds are growing over time, apart from outliers and the hail quantity, their main value distributions do not change significantly. Correlations of initial parameters with the distances to the median member were found. A more detailed analysis based on different reference members and revealing higher-dimensional dependencies to initial parameters is planned for future work. Further, the application of the workflow on other data sets could lead to interesting insights.

Acknowledgements

This research has been done within the subproject B5 of the Transregional Collaborative Research Center SFB/TRR 165 Waves to Weather funded by the German Research Foundation (DFG).

References

[ASE15] ATHAWALE, T., SAKHAEE, E., and ENTEZARI, A. "Isosurface visualization of data with nonparametric models for uncertainty". *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2015), 777–786 3.

[BM10] BRUCKNER, S. and MOLLER, T. "Result-driven exploration of simulation parameter spaces for visual effects design". *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), 1468–1476 3.

[BvLBS11] BREMM, S., von LANDESBERGER, T., BERNARD, J., and SCHRECK, T. "Assisted Descriptor Selection Based on Visual Comparative Data Analysis". *Computer Graphics Forum* 30.3 (2011), 891–900 3.

[DDW14] DEMIR, I., DICK, C., and WESTERMANN, R. "Multi-Charts for Comparative 3D Ensemble Visualization". *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014) 3.

[DJW16] DEMIR, I., JAREMA, M., and WESTERMANN, R. "Visualizing the Central Tendency of Ensembles of Shapes". *SIGGRAPH Asia 2016 Symposium on Visualization*. SA '16. ACM, 2016 3.

[DS15] DUTTA, S. and SHEN, H.-W. "Distribution driven extraction and tracking of features for time-varying data analysis". *IEEE transactions on visualization and computer graphics* 22.1 (2015), 837–846 3.

[EKS*96] ESTER, M., KRIEGEL, H.-P., SANDER, J., XU, X., et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." *Kdd*. Vol. 96. 34. 1996, 226–231 2, 4.

[FBW16] FERSTL, F., BÜRGER, K., and WESTERMANN, R. "Streamline Variability Plots for Characterizing the Uncertainty in Vector Field Ensembles". *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), 767–776 3.

[FJ05] FRED, A. L. N. and JAIN, A. K. "Combining multiple clusterings using evidence accumulation". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.6 (June 2005), 835–850 3, 4.

[FKRW16] FERSTL, F., KANZLER, M., RAUTENHAUS, M., and WESTERMANN, R. "Visual Analysis of Spatial Variability and Global Correlations in Ensembles of Iso-Contours". *Computer Graphics Forum* 35.3 (2016), 221–230 3.

[FKRW17] FERSTL, F., KANZLER, M., RAUTENHAUS, M., and WESTERMANN, R. "Time-hierarchical Clustering and Visualization of Weather Forecast Ensembles". *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), 831–840 3.

[FT74] FRIEDMAN, J. H. and TUKEY, J. W. "A Projection Pursuit Algorithm for Exploratory Data Analysis". *IEEE Transactions on Computers* C-23.9 (Sept. 1974), 881–890 3.

[HA85] HUBERT, L. and ARABIE, P. "Comparing partitions". *Journal of classification* 2.1 (1985), 193–218 3.

[HBS17] HAZARIKA, S., BISWAS, A., and SHEN, H.-W. "Uncertainty visualization using copula-based analysis in mixed distribution models". *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017), 934–943 3.

[HDSC19] HAZARIKA, S., DUTTA, S., SHEN, H., and CHEN, J. "CoDDA: A Flexible Copula-based Distribution Driven Analysis Framework for Large-Scale Multivariate Data". *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), 1214–1224 3, 5.

[HOGJ13] HUMMEL, M., OBERMAIER, H., GARTH, C., and JOY, K. I. "Comparative visual analysis of Lagrangian transport in CFD ensembles". *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), 2743–2752 3.

[JDKW15] JAREMA, M., DEMIR, I., KEHRER, J., and WESTERMANN, R. "Comparative visual analysis of vector field ensembles". *IEEE Conference on Visual Analytics Science and Technology (VAST)*. 2015, 81–88 3.

[JHB*17] JÄCKLE, D., HUND, M., BEHRISCH, M., et al. "Pattern Trails : Visual Analysis of Pattern Transitions in Subspaces". *IEEE Conference on Visual Analytics Science and Technology (VAST)*. 2017, 1–12 3.

[KH13] KEHRER, J. and HAUSER, H. "Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey". *IEEE Transactions on Visualization and Computer Graphics* 19.3 (Mar. 2013), 495–513 2.

[KRRW19] KUMPF, A., RAUTENHAUS, M., RIEMER, M., and WESTERMANN, R. "Visual Analysis of the Temporal Evolution of Ensemble Forecast Sensitivities". *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), 98–108 4, 5.

[KTB*18] KUMPF, A., TOST, B., BAUMGART, M., et al. "Visualizing Confidence in Cluster-based Ensemble Weather Forecast Analyses". *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), 109–119 3.

[Kuh55] KUHN, H. W. "The Hungarian method for the assignment problem". *Naval research logistics quarterly* 2.1-2 (1955), 83–97 5, 6.

[KW78] KRUSKAL, J. B. and WISH, M. *Multidimensional scaling*. Vol. 11. Sage, 1978 4.

[LLBP12] LIU, S., LEVINE, J. A., BREMER, P.-T., and PASCUCCI, V. "Gaussian mixture model based volume visualization". *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*. IEEE. 2012, 73–77 3.

[LMW*16] LIU, S., MALJOVEC, D., WANG, B., et al. "Visualizing high-dimensional data: Advances in the past decade". *IEEE Transactions on Visualization and Computer Graphics* 23.3 (2016), 1249–1268 2.

[LPK05] LOVE, A. L., PANG, A., and KAO, D. L. "Visualizing spatial multivalue data". *IEEE Computer Graphics and Applications* 25.3 (2005), 69–79 3.

[LSP*10] LEX, A., STREIT, M., PARTL, C., et al. "Comparative Analysis of Multidimensional, Quantitative Data". *IEEE Transactions on Visualization and Computer Graphics* 16.6 (Nov. 2010), 1027–1035 3.

[LT16] LEHMANN, D. J. and THEISEL, H. "Optimal Sets of Projections of High-Dimensional Data". *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), 609–618 3.

[MH08] MAATEN, L. V. D. and HINTON, G. "Visualizing data using t-SNE". *Journal of Machine Learning Research* 9 (2008), 2579–2605 2.

[MWK14] MIRZARGAR, M., WHITAKER, R. T., and KIRBY, R. M. "Curve boxplot: Generalization of boxplot for ensembles of curves". *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), 2654–2663 3.

[OLK*14] OELTZE, S., LEHMANN, D. J., KUHN, A., et al. "Blood flow clustering and applications in virtual stenting of intracranial aneurysms". *IEEE Transactions on Visualization and Computer Graphics* 20.5 (2014), 686–701 3.

[PH13] PÖTHKOW, K. and HEGE, H.-C. "Nonparametric models for uncertainty visualization". *Computer Graphics Forum*. Vol. 32. 3pt2. Wiley Online Library. 2013, 131–140 3.

[PHL*16] PEZZOTTI, N., HÖLLT, T., LELIEVELDT, B., et al. "Hierarchical Stochastic Neighbor Embedding". *Computer Graphics Forum* 35.3 (2016), 21–30 4.

[SG02] STREHL, A. and GHOSH, J. "Cluster ensembles—a knowledge reuse framework for combining multiple partitions". *Journal of machine learning research* 3.Dec (2002), 583–617 3.

[VR11] VEGA-PONS, S. and RUIZ-SHULCLOPER, J. "A survey of clustering ensemble algorithms". *International Journal of Pattern Recognition and Artificial Intelligence* 25.03 (2011), 337–372 3.

[WBJ*18] WELLMANN, C., BARRETT, A., JOHNSON, J., et al. "Using Emulators to Understand the Sensitivity of Deep Convective Clouds and Hail to Environmental Conditions". *Journal of Advances in Modeling Earth Systems* 10.12 (2018), 3103–3122 2, 3.

[WCR*18] WENSKOVITCH, J., CRANDELL, I., RAMAKRISHNAN, N., et al. "Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics". *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), 131–141 3.

[WLW*17] WANG, K.-C., LU, K., WEI, T.-H., et al. "Statistical visualization and analysis of large data using a value-based spatial distribution". *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE. 2017, 161–170 3.

[WMK13] WHITAKER, R. T., MIRZARGAR, M., and KIRBY, R. M. "Contour Boxplots: A Method for Characterizing Uncertainty in Feature Sets from Simulation Ensembles". *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), 2713–2722 3.