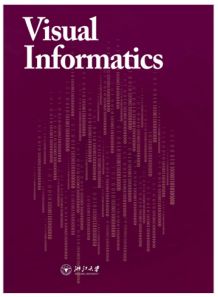
Journal Pre-proof



CECAV-DNN: Collective Ensemble Comparison and Visualization using Deep Neural Networks

Wenbin He, Junpeng Wang, Hanqi Guo, Han-Wei Shen, Tom Peterka

PII: S2468-502X(20)30016-4
DOI: https://doi.org/10.1016/j.visinf.2020.04.004
Reference: VISINF 68

To appear in: *Visual Informatics*

Please cite this article as: W. He, J. Wang, H. Guo et al., CECAV-DNN: Collective Ensemble Comparison and Visualization using Deep Neural Networks. *Visual Informatics* (2020), doi: https://doi.org/10.1016/j.visinf.2020.04.004.

Visual Informatics (2020)

Contents lists available at ScienceDirect

Visual Informatics

journal homepage: www.elsevier.com/locate/visinf

CECAV-DNN: Collective Ensemble Comparison and Visualization using Deep Neural Networks

Wenbin He[a], Junpeng Wang[b], Hanqi Guo[c], Han-Wei Shen[b], Tom Peterka[c]

[a]*The Ohio State University, Columbus, Ohio, United States*
[b]*Visa Research, Palo Alto, California, United States*
[c]*Argonne National Laboratory, Lemont, Illinois, United States*

ARTICLE INFO

Article history:
Received 1 May 2013
Received in final form 10 May 2013
Accepted 13 May 2013

Keywords: Collective ensemble comparison, ensemble data visualization, deep neural networks

ABSTRACT

We propose a deep learning approach to collectively compare two or multiple ensembles, each of which is a collection of simulation outputs. The purpose of collective comparison is to help scientists understand differences between simulation models by comparing their ensemble simulation outputs. However, the collective comparison is non-trivial because the spatiotemporal distributions of ensemble simulation outputs reside in a very high dimensional space. To this end, we choose to train a deep discriminative neural network to measure the dissimilarity between two given ensembles, and to identify when and where the two ensembles are different. We also design and develop a visualization system to help users understand the collective comparison results based on the discriminative network. We demonstrate the effectiveness of our approach with two real-world applications, including the ensemble comparison of the community atmosphere model (CAM) and the rapid radiative transfer model for general circulation models (RRTMG) for climate research, and the comparison of computational fluid dynamics (CFD) ensembles with different spatial resolutions.

1. Introduction

Over the last decade, ensemble simulations have been playing an increasingly important role in various scientific and engineering disciplines, such as computational fluid dynamics (CFD), aerodynamics, climate, and weather research. Scientists routinely conduct a set of simulations with different parameters, and study the sensitivities of the simulation models by comparing outputs of individual runs, namely ensemble members or simply *members*.

This study focuses on the *collective comparison* of members. For ease of description, we use the term *ensemble* as a group of members. Instead of making comparisons between individual members, we aim to collectively compare different groups of members in this study. The demand for collective ensemble comparison arises from many real-world applications. For example, in developing a new simulation model, scientists often interested in how the model performs compared with old models such that scientists can have a better understanding of the new model and eventually improve it. Scientists sometimes also need to compare simulation models with different spatial resolutions to select a spatial resolution that balances accuracy and computation cost.

Based on our discussion with domain scientists, the purpose of collective comparison between any two ensembles is to answer three specific questions: (1) How to measure the overall dissimilarity between the two ensembles? In other words, what is the dissimilarity between the two distributions of simulation outputs? (2) Which members of the two ensembles agree or disagree with each other? The agreement or disagreement between members depends not only on the similarities between them but also on their probabilities of occurrence in the two distributions

e-mail: he.495@osu.edu (Wenbin He), junpenwa@visa.com (Junpeng Wang), hguo@anl.gov (Hanqi Guo), shen.94@osu.edu (Han-Wei Shen), tpeterka@anl.gov (Tom Peterka)

of simulation outputs. (3) What spatial regions are the most important to differentiate the two ensembles?

The collective comparison between ensembles has not been well developed. Although ensemble visualization has been extensively studied recently, the majority of previous ensemble visualization techniques focused on comparing individual members within an ensemble. Several pioneering works in comparing multiple ensembles include: (1) the analysis of simulation parameters for ensembles with different spatial resolutions (Wang et al., 2017; Biswas et al., 2017), where the focus is on parameter analysis; (2) visual comparison of multiple collections/ensembles of scalar values (Höllt et al., 2014) or 2D isocontours (Ferstl et al., 2017) via juxtaposition or superimposition, where the focus is on qualitative comparison of particular features. However, effective visualization and comprehensive analysis of variability between ensembles of simulation outputs remain challenging.

The key research challenge of collective ensemble comparison is to measure the divergence of two given ensembles. The metric must incorporate all values of the simulation solution, such as values on every grid points, which resides in a high dimensional space. A straightforward approach is to model the probability density function (PDF) of each ensemble, and then measure the divergence between the PDFs. However, the dimensionality of the simulation solution space is usually prohibitively high for modeling and comparing PDFs. Even if we measured the divergence of two PDFs, it is still non-trivial to visualize and analyze where the two PDFs are different (questions (2) and (3)).

We explore to use deep neural networks (DNNs) to tackle the challenge of measuring the divergence of high-dimensional distributions for collective ensemble comparison. DNN is a emerging technology that makes it possible to model and compare distributions of images (e.g. generative models (Goodfellow et al., 2014; Nowozin et al., 2016; Arjovsky et al., 2017; Gulrajani et al., 2017; Radford et al., 2015)). Without modeling a PDF for a distribution of images defined on high dimensional space, DNN approaches can differentiate images of one distribution (e.g., fake images) from images of the other distribution (e.g., true images), and also measure the dissimilarity between the two distributions. Similarly, we believe DNN approaches are also promising to visualize and analyze the difference between two ensembles.

In this study, we propose an approach to perform Collective Ensemble Comparison And Visualization using Deep discriminative Neural Networks. A discriminative network is a specific type of DNN, which can compare two probability distributions represented by two sets of samples. In our context, a discriminative network is trained to differentiate members of one ensemble from members of the other. Through training, each individual member (from both ensembles) will be assigned a numerical score by the discriminative network, indicating the likelihood that the member is from one ensemble rather than the other. By analyzing and visualizing the outputs of the discriminative network, our approach provides three-level comparative analysis of the two ensembles to answer the aforementioned questions: (1) We analyze the loss value of the discriminative network to measure the overall dissimilarity between the two ensembles. (2) By visualizing and analyzing the distributions of the two collections of likelihood scores, the agreement and disagreement between individual members of the two ensembles can be studied. (3) Using the internal parameters of the discriminative network, the importance of different spatial locations in differentiating the two ensembles can be characterized. Furthermore, to support the three-level comparative analysis mentioned above, we design and develop a visualization system based on the outputs of the discriminative network. Our system supports not only the three-level comparison of a single pair of ensembles, but also the comparison among multiple pairs of ensembles simultaneously (e.g., a temporal sequence of ensemble pairs).

We demonstrate the effectiveness and usefulness of the proposed approach using two real-world use-cases, and verify the results with a domain scientist from environmental sciences. In the first case, we compare ensembles generated using different weather forecast models in climate research (Yang et al., 2012). In the second case, we study the influence of spatial resolutions on the outputs of ensemble simulations in the field of fluid dynamics. With the proposed approach, the difference between ensembles can be identified and visualized effectively. In summary, the contributions of this study are twofold:

- A deep learning approach (CECAV-DNN) for collective comparison and visualization of multiple ensembles. Discriminative networks are trained to identify how and where the ensembles are different
- A visualization system to explore and analyze the outputs of discriminative networks with respect to the difference between ensembles

2. Related Work

In this section, we summarize related work on ensemble visualization and distribution comparison.

2.1. Ensemble Visualization

There are two distinguished lines of research topics in ensemble visualization, namely individual and collective ensemble comparison. The former aims at comparing individual members within an ensemble, and the latter makes comparison between ensembles.

Individual Ensemble Comparison The majority of existing ensemble visualization techniques focused on analysis and visualization of variability among members of an ensemble, which can be classified into location-based and feature-based techniques (Obermaier and Joy, 2014). Location-based techniques compare scalar or vector values of ensemble members at fixed locations and visualize the variability of the members using pseudo-coloring (Potter et al., 2009; Hummel et al., 2013; Gosink et al., 2013; Bensema et al., 2016; Sakhaee and Entezari, 2017) or glyphs (Sanyal et al., 2010; Kehrer et al., 2011; Jarema et al., 2015; Hlawatsch et al., 2011). Feature-based techniques first extract features (e.g., isocontours, streamlines) from individual members and then compare them across the

ensemble. For the visualization of feature variability in ensemble datasets, two major groups of techniques exist. The first group of feature-based techniques composites visualization of features extracted from individual members using spaghetti plot (Diggle et al., 2013; Potter et al., 2009), surface slicing (Alabi et al., 2012), screen space silhouettes (Demir et al., 2016), or screen door tinting (Phadke et al., 2012). The second group of feature-based techniques first derives summary statistics (e.g., order statistics (Whitaker et al., 2013; Mirzargar et al., 2014; Raj et al., 2016), level crossing probabilities (Pöthkow et al., 2011; Pöthkow and Hege, 2011, 2013; Athawale and Entezari, 2013; Athawale et al., 2016), clusters (Ferstl et al., 2016a,b, 2017; Kumpf et al., 2018), density estimates (Guo et al., 2016)) of the extracted features, and then encodes the derived summary statistics into visualizations. Compared with these techniques, our work focuses on analyzing variability between different ensembles instead of members within an ensemble.

Collective Ensemble Comparison Comparisons between ensembles play an important role in analyzing various simulation models, investigating different spatial resolutions, and exploring the temporal evolution of ensembles. A few techniques have been proposed to tackle the problem of collective ensemble comparison, and much work remains to be done. Existing techniques include comparison between ensembles of scalar values, simulation parameters, and isocontours.

Höllt et al. (2014) compared two ensembles of scalar values at different spatial locations across time by visualizing the distributions of the two ensembles of scalar values side-by-side at each timestep. Köthur et al. (2015) extended the use of the windowed cross-correlation matrix to support a correlation-based comparison between two ensembles of time series. However, the methods of Höllt et al. and Köthur et al. are limited to ensembles of scalar values and not feasible for the comparison between ensembles of scalar fields.

Wang et al. (2017) and Biswas et al. (2017) proposed methods to investigate climate ensembles of different spatial resolutions for the analysis of the simulation parameters and the prediction accuracy. Wang et al. introduced the nested parallel coordinates plot to visualize intra-resolution and inter-resolution parameter correlations. Biswas et al. analyzed and visualized the sensitivity and accuracy of the ensembles with respect to the simulation parameters across different spatial resolutions. However, the methods of Wang et al. and Biswas et al. mainly focused on investigating the influence of different spatial resolutions on the sensitivity of the simulation parameters and the accuracy of the prediction results compared with the observed ground truth. Comparisons between ensembles of simulation fields to investigate where the ensembles agree or disagree with each other are missing.

Pfaffelmoser and Westermann (2013) proposed a technique to indicate the difference between ensembles of isocontours using spaghetti plots colored by gradients along the isocontours and backgrounds colored by probabilities of scalar values greater than the isovalue. Ferstl et al. (2017) analyzed ensembles of 2D isocontours at different timesteps using time-hierarchical clustering and visualized the temporal evolution of the clusters using stacked contour variability plots. However, the methods of Pfaffelmoser et al. and Ferstl et al. mainly focused on visual comparison between two collections of 2D isocontours. Collective comparison and visualization of the difference between distributions of simulation outputs remain challenging.

2.2. Distribution Comparison

The collective ensemble comparison is related to distribution comparison. In our work, ensemble members are considered as samples that model a distribution defined in high dimensional space. We compare any two ensembles by comparing the distributions that modeled by the samples, and extract members and spatial regions that are the most important for differentiating the two distributions. Comparison between distributions are widely used in our community for various applications, such as feature searching (Wei et al., 2015, 2017; Dutta et al., 2017a,b; Hazarika et al., 2018) and tracking (Dutta and Shen, 2016), streamline similarity analysis (Lu et al., 2013), clustering (He et al., 2017), and dimensionality reduction (Chen et al., 2015).

Various techniques have been proposed to analyze the difference between two distributions based on a collection of samples from each distribution in the fields of statistics, information theory, and machine learning. A straightforward approach is to model a PDF (e.g. histogram) for each distribution and compare the difference between the two PDFs. However, modeling and comparing high dimensional distributions are challenging. A series of k-nearest-neighbor based approaches (Wang et al., 2009; Poczos and Schneider, 2011; Moon and Hero, 2014; Póczos et al., 2012) have been proposed to measure the divergence between two multidimensional distributions based on samples. However, nearest neighbor search becomes unreliable in a high dimensional space because of the sparsity of the samples. Maximum mean discrepancy (MMD) has been proposed in (Chwialkowski et al., 2015; Jitkrittum et al., 2016; Gretton et al., 2007) to measure the distance between two sets of samples and identify when and where the two underlying distributions are different. However, many frequently used divergences (e.g., Kullback-Leibler divergence and earth mover's distance) are not supported by MMD based methods. Recently, deep discriminative neural networks have been playing an increasingly important role in comparing high dimensional distributions represented by samples using various divergence/distance between distributions. For example, generative adversarial networks (GANs) (Goodfellow et al., 2014; Nowozin et al., 2016; Arjovsky et al., 2017; Gulrajani et al., 2017; Radford et al., 2015) used discriminative neural networks to estimate divergences between real and fake images and updated the weights of networks based on the estimated divergences. Lopez-Paz and Oquab (2016) performed two-sample tests based on discriminative neural networks. Im et al. (2018) used discriminative neural networks to evaluate the performance of generative models. In this work, discriminative networks are used to perform collective comparison between ensembles, as detailed in following sections.

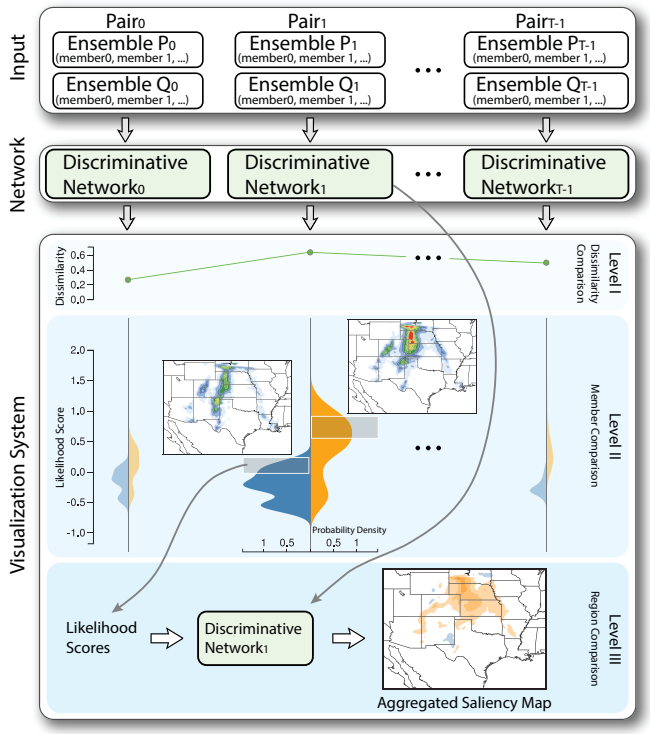


Fig. 1. Workflow of CECV, which takes a sequence of ensemble pairs as input, and trains a discriminative network for each ensemble pair. After training, three levels of comparative analysis are provided with an interactive visualization system to compare the ensemble pairs.

3. Overview

Figure 1 shows the workflow of CECV to collectively compare multiple ensembles. The input of our approach is a sequence of ensemble pairs (e.g., a temporal sequence of ensemble pairs generated with different simulation models), where each ensemble is a collection of members (i.e. scalar fields). We first train a sequence of discriminative networks to perform comparative analysis on the ensemble pairs. Then, we design and develop a visualization system to facilitate the comparative analysis based on the trained discriminative networks.

We train a discriminative network to differentiate each pair of ensembles. After training, the discriminative network assigns a likelihood score to each member, which indicates the likelihood that the member is from one ensemble rather than the other. Based on the outputs of the trained discriminative network, our approach provides three levels of comparative analysis. First, we measure the dissimilarity between the two ensembles based on the loss value of the discriminative network. Second, we compare the distributions of the two collections of likelihood scores to identify members in which the two ensembles agree or disagree with each other. Third, by taking advantage of the back-propagation algorithm (Rumelhart et al., 1986), the spatial regions that are the most sensitive in differentiating the two ensembles are identified.

We design and develop an interactive visualization system to facilitate the analysis on the trained discriminative networks, which empowers the three-level comparative analysis mentioned above (details in Section 5). For the first level analysis, the overall dissimilarities are visualized with line charts, which provides informative hints to help users focus on a particular pair in the sequence for further exploration (e.g., the pair of ensembles at a timestep with the maximum dissimilarity). A sequence of violin plots are used to encode and compare the distributions of likelihood scores for the second level analysis. For each violin plot, users can explore different sub-ranges of the PDF via brushing, and visualize the corresponding members as well as the sensitive spatial regions for the third level analysis.

4. Collective Ensemble Comparison

Our method starts with a pair of ensembles $P = \{p_0, p_1, \ldots, p_{n-1}\}$ and $Q = \{q_0, q_1, \ldots, q_{m-1}\}$. Each member in the ensembles, p_i or q_i, is a scalar field, and values at different locations of the field denote the simulation output of a certain variable (e.g. temperature, precipitation) from the corresponding ensemble run. Members in individual ensembles share the same spatial region (i.e. same mesh discretization). Mathematically, we consider each member p_i as a high-dimensional vector in $\mathbb{R}^M$, where M is the number of grid points in the spatial field. Then, the two ensembles P and Q are considered as two sets of samples that are sampled from two probability distributions defined on $\mathbb{R}^M$.

Inspired by the recent advances of generative models (Goodfellow et al., 2014; Nowozin et al., 2016; Arjovsky et al., 2017; Gulrajani et al., 2017; Radford et al., 2015), which use a discriminative network to compare two probability distributions represented by two sets of samples, we train a discriminative network to quantify the dissimilarity between the two ensembles and identify members (or spatial regions of those members) in which the two ensembles are different. Specifically, our discriminative network differentiates members of one ensemble (P) from members of the other (Q) by assigning each member a likelihood score, such that the difference between the two collections of likelihood scores for members in P and Q is maximized. In this way, we transform the problem of comparing two ensembles of vectors in $\mathbb{R}^M$ into comparing two collections of likelihood scores in $\mathbb{R}$.

In the rest of this section, we first provide foundations for our neural network based approach, and then elaborate the three-level comparative analysis between a single pair of ensembles. We discuss how our approach can be extended to compare multiple pairs of ensembles, which are often required in real-world applications.

4.1. Discriminative Networks

In this section, we first describe the basic concepts in the architecture of discriminative networks, and then discuss the objective function and the training process of discriminative networks.

4.1.1. Architecture

A discriminative network $\mathcal{D}$ (shown in Figure 2) is a nonlinear function. This function maps a member, i.e., a high-dimensional vector, in $\mathbb{R}^M$ to a likelihood score in $\mathbb{R}$. Specifi-

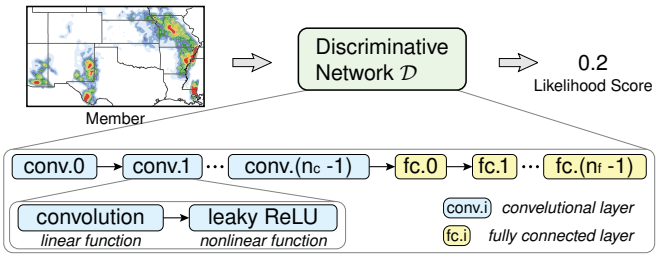


Fig. 2. Architecture of discriminative network $\mathcal{D}$, which begins with a sequence of convolutional layers, followed by a few fully connected layers.

cally, $\mathcal{D}$ is implemented through an alternating sequence of linear and nonlinear functions (i.e. activation functions), where each pair of the linear and nonlinear function is commonly referred to as a *layer* of the network. Based on the linear computations in the function pair, two types of layers are commonly used in $\mathcal{D}$: convolutional layer and fully connected layer.

Convolutional Layers Convolutional layers perform linear convolutions to extract features from the input. Because we work with scientific data that has spatial continuities, our $\mathcal{D}$ starts with a sequence of such computations to elicit features layer by layer. Each convolutional layer consists of many trainable *filters* (i.e. convolutional kernels), and each filter can extract a specific type of spatial features from the input. Also, because the convolution operations are performed with a fixed stride, the output of each convolutional layer is usually a downsampled version of the input, in which certain features are highlighted.

Fully Connected Layers The fully connected layers of our discriminative network reduce the features maps from the last convolutional layer to a numerical value. Each fully connected layer performs a matrix multiplication to assign weights to different elements of the input and aggregate those elements. To the last fully connected layer, the original input (outputs from the last convolutional layer) is reduced to a numerical value (i.e. a likelihood score).

Activation Function The output from both convolutional and fully connected layers will be fed into a nonlinear activation function to filter out inactivate elements from the outputs. Same as the generative models (Arjovsky et al., 2017; Gulrajani et al., 2017; Radford et al., 2015), we use Leaky Rectified Linear Units (ReLUs) as the activation function for all layers of $\mathcal{D}$, which is defined as

$$a(x) = \begin{cases} x, & \text{if } x > 0 \\ -cx, & \text{otherwise} \end{cases}, \tag{1}$$

where c is a constant value, and typically set to 0.2 in $\mathcal{D}$ (Radford et al., 2015).

The trainable parameters of $\mathcal{D}$ are the filters in convolutional layers and the weights in fully connected layers. In this work, we denote a discriminative network defined by a collection of parameters ϕ as $\mathcal{D}_\phi$. To train a discriminative network $\mathcal{D}_\phi$ that can differentiate two given ensembles, the parameters ϕ need to be optimized based on an objective function through an iterative training process, which are detailed in the following sections.

4.1.2. Objective Function

Given two ensembles P and Q, we map them into two collections of likelihood scores through $\mathcal{D}_\phi$, and use an objective function to optimize the mapping, so that the difference between the two collections of likelihood scores is maximized. More importantly, from the resulting collections of likelihood scores and the optimized parameters ϕ, we are able to detect members in which, and spatial regions where, the two ensembles are different.

Various objective functions (Goodfellow et al., 2014; Nowozin et al., 2016; Arjovsky et al., 2017; Gulrajani et al., 2017) have been proposed for the training of a discriminative network to differentiate two probability distributions represented by two sets of samples. The maximum value of a specific objective function usually reflects the distance or divergence between the two distributions. In this work, we demonstrate our approach using the objective function proposed in (Arjovsky et al., 2017; Gulrajani et al., 2017), which has a maximum value corresponds to the Wasserstein distance (i.e. earth mover's distance) (Müller, 1997). The Wasserstein distance can be used to measure the dissimilarity between distributions that are not overlapping with each other, which is important for high dimensional distributions as they are often sparse. By treating two ensembles P and Q as two sets of samples from two probability distributions, the Wasserstein distance $W(P, Q)$ between the two distribution is defined as:

$$W(P, Q) = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{p \sim P}[\, f(p) \,] - \mathbb{E}_{q \sim Q}[\, f(q) \,] \right), \tag{2}$$

where p and q are members sampled from P and Q, respectively, $\mathbb{E}(\cdot)$ represents expectation, f is a 1-Lipschitz function (i.e. a function that satisfies $|f(x) - f(y)| \leq |x - y|$ for all x and y) mapping from $\mathbb{R}^M$ to $\mathbb{R}$, $\mathcal{F}$ is a class of functions that are all 1-Lipschitz. By limiting $\mathcal{D}_\phi$ as a parameterized family of functions that are all 1-Lipschitz (details in (Arjovsky et al., 2017)), the objective function L is defined as:

$$L(P, Q; \phi) = \mathbb{E}_{p \sim P}[\, \mathcal{D}_\phi(p) \,] - \mathbb{E}_{q \sim Q}[\, \mathcal{D}_\phi(q) \,]. \tag{3}$$

Through iteratively maximizing the objective function, the Wasserstein distance between P and Q can be measured.

Although we focus on the objective function corresponding to the Wasserstein distance in our study, the proposed method is flexible to use objective functions corresponding to other widely used distances or divergences (e.g., Kullback-Leibler divergence), which will be discussed in Section 8.

4.1.3. Training Process

As shown in Algorithm 1, we discuss the training process used to optimize the parameters ϕ of a discriminative network $\mathcal{D}_\phi$ with respect to the objective function $L(P, Q; \phi)$. We use stochastic gradient descent (Bottou, 2010) to iteratively optimize the parameters ϕ, as shown in Algorithm 1. In each iteration, a batch of members are sampled from each of the two ensembles randomly and fed into the neural network (lines 2–3). Then, the gradient of the objective function is computed with respect to the current parameters ϕ using back-propagation (Rumelhart et al., 1986), which computes gradients from output to input layer by layer (i.e., starting from the

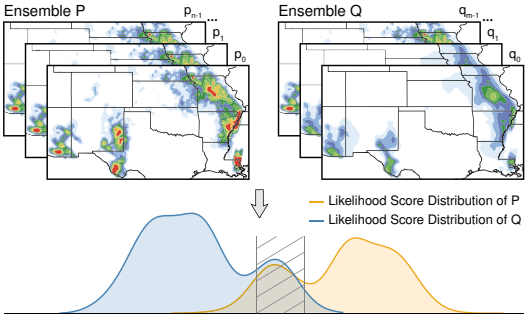


Fig. 3. Comparing two ensembles using the two distributions of likelihood scores produced by the trained discriminative network $\mathcal{D}_\phi$.

last layer and propagating back to the first layer). Note that we introduce a gradient penalty gp to the loss function to enforce the Lipschitz-continuity of $\mathcal{D}_\phi$ as presented in (Gulrajani et al., 2017). The gradient penalty is defined as the gradient norm of the network's output with respect to the weighted combination of members from one ensemble and the other as $(\| \nabla \mathcal{D}_\phi(\epsilon P' + (1-\epsilon)Q') \|_2 - 1)^2$, where the weight ϵ is sampled from 0 to 1 randomly. Based on the resulting gradients, the parameters ϕ are updated (line 4) using the Adam optimizer with default settings (i.e. $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$) (Kingma and Ba, 2014), the one that has been widely used in the training of discriminative networks. We keep executing the loop (lines 1–5) of updating the parameters ϕ until the exit criteria is satisfied (e.g. reaching the maximum number of iterations).

Algorithm 1 Training process of the discriminative network $\mathcal{D}_\phi$. Parameters ϕ are initialized by sampling from a Gaussian distribution randomly. b is the batch size. The function `random_sample`(P, b) randomly samples b members from the input ensemble P. L is the objective function. gp is the gradient penalty used to enforce the Lipschitz-continuity of $\mathcal{D}_\phi$ as presented in (Gulrajani et al., 2017), and λ is a constant weight set to 10. α, β_1, β_2 are the parameters of the Adam optimizer (Kingma and Ba, 2014). $\nabla_\phi L(P', Q'; \phi)$ is the gradient of L with respect to ϕ.

Input: Initial parameters ϕ of the discriminative network, ensembles P and Q
Output: Optimized parameters ϕ
1: Repeat:
2: $P' \leftarrow$ `random_sample`(P, b)
3: $Q' \leftarrow$ `random_sample`(Q, b)
4: $\phi \leftarrow \text{Adam}(\nabla_\phi L(P', Q'; \phi) + \lambda \times gp, \phi, \alpha, \beta_1, \beta_2)$
5: Until exit criteria is satisfied

4.2. Three-Level Comparative Analysis of Two Ensembles

In this section, we explain how we use the trained discriminative network $\mathcal{D}_\phi$ to collectively compare a pair of ensembles. By analyzing and visualizing the result of the objective function, the two collections of likelihood scores produced by $\mathcal{D}_\phi$, and the network parameters ϕ, our approach provides three levels of comparative analysis between the two ensembles, which are detailed as follows.

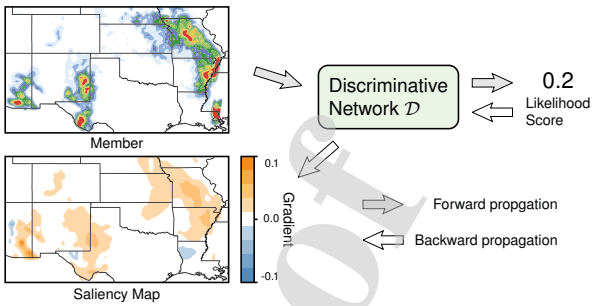


Fig. 4. Saliency map of a member computed using backpropagation.

Level I: Dissimilarity Comparison This level measures the overall dissimilarity between two given ensembles. It is conducted by optimizing the objective function (Equation 3) of $\mathcal{D}_\phi$ for the two ensembles. To the end, the maximum value of the objective function can measure the distance between the two probability distributions represented by the two sets of samples (i.e., the two input ensembles).

Level II: Member Comparison This level identifies the members in which the two simulations agree or disagree with each other. In other words, it compares the probabilities of occurrence for individual members in the two ensembles. The output of $\mathcal{D}_\phi$, i.e., a likelihood score, indicates the likelihood that the input member is from one ensemble rather than the other. By mapping the two ensembles to two collections of likelihood scores through $\mathcal{D}_\phi$ (Figure 3) and modeling them to two distributions, we can more effectively compare the original high-dimensional members in $\mathbb{R}^M$. For example, for the ranges (of the likelihood scores) where the two distributions overlap (the shaded area in Figure 3), the corresponding members are the common trend of the two ensembles, which means the simulations agree with each other in those members. On the other hand, for the ranges that the two distributions do not overlap, the corresponding members are highly different for the two simulations, which could be further investigated to understand the difference between the simulations.

Level III: Region Comparison This level extracts spatial regions, which are sensitive to differentiate the two ensembles (Figure 4). It is conducted by computing how strong each spatial location of each member is affecting the likelihood score from $\mathcal{D}_\phi$. Specifically, one member $p_i \in \mathbb{R}^M$ can be denoted as $p_i = (s_0, s_1, \ldots s_{M-1})$, where s_j ($j \in [0, M-1]$) is the value associated with the jth spatial location (grid point) of the member; and the likelihood score for this member is $\mathcal{D}_\phi(p_i)$. By computing the gradient of $\mathcal{D}_\phi(p_i)$ with respect to each s_j, i.e. $\frac{\partial \mathcal{D}_\phi}{\partial s_j}$, using backpropagation (Rumelhart et al., 1986), we can disclose the influence of s_j on the likelihood score of p_i from two aspects:

- The magnitude of the gradient measures the sensitivity of s_j with respect to the likelihood score. When the gradient is high, it means a little change of s_j affects the likelihood score of p_i strongly;
- The sign of the gradient indicates that when s_j increases, the member p_i is more likely to come from one ensemble or the other.

Extending the computation of the gradient to all $s_j \in p_i$, we derive a saliency map (Simonyan et al., 2013) for p_i, which has the same size with p_i and highlights the spatial locations whose values are the most important for differentiating the two ensembles, as shown in Figure 4.

4.3. Extending to Multiple Pairs of Ensembles

We noticed that comparisons are often needed among multiple pairs of ensembles in real-world applications. For example, in order to analyze two spatiotemporal climate ensembles, scientists often need to compare the two ensembles at individual timesteps (i.e., a sequence of ensemble pairs). Our proposed three-level comparative analysis between a single pair of ensembles can easily be extended to multiple pairs. For example, the single numerical value in level one will become a sequence of numerical values and the two probability distributions in level two will become a sequence of distribution pairs.

As we extending the analysis scope (from one pair to multiple pairs of ensembles) to cope with real-world applications, we realize the need of a visualization system. The visualization system can help us better organize different facets of the complicated ensemble data and facilitate our three-level comparative analysis. Moreover, a process of visual exploration emerges to be necessary to avoid information overload. For example, when facing with two large temporal sequence of ensembles, users should be able to get an effective overview of them first, and the overview should provide informative guidance to users' further detailed investigations. Friendly user interactions to help flexibly switch between the overview and details would significantly improve the exploration experience. In an attempt to address these requirements, we design a visualization system to support our three-level comparative analysis for multiple pairs of ensembles. The details of the system is further explained in Section 5.

5. Visualization System for Collective Comparison

We design and develop an visualization system to help domain scientists compare ensembles collectively. The visualization system support the three-level comparative analysis by visualizing and analyzing the outputs of the trained discriminative networks. In this section, we describe the design considerations and choices of our interface, and provide guidelines for visual exploration using our system.

5.1. User Interface

The proposed visualization system is constituted of three coordinated views: the parallel violins plot (PVP) view, the member view, and the saliency map view. Details for each view are discussed as follows.

5.1.1. Parallel Violins Plot (PVP) View

The PVP view (Figure 5(a)) consists of a dissimilarity chart and a sequence of violin plots, which presents the overall dissimilarities and the distributions of likelihood scores for a sequence of ensemble pairs that users are interested in. The PVP view is used as user interface to support our level I and level II comparative analysis.

Dissimilarity Chart A dissimilarity chart is a line chart to present the sequence of dissimilarities for the sequence of ensemble pairs, as shown in Figure 5(a1). This straightforward visualization enables users to quickly identify interesting pairs (e.g. ensemble pairs with larger dissimilarities) for further detailed investigations.

Violin Plots We use the violin plots (Hintze and Nelson, 1998; Höllt et al., 2014) to encode and compare the distributions of likelihood scores. For each pair of ensembles, we first transform them into two collections of likelihood scores using the trained discriminative network. Then, we model a 1D PDF for each collection of likelihood scores. In order to visually compare the two PDFs, we joint them into a violin plot by putting them as side-by-side views for juxtaposed comparison, as shown in Figure 6(a). For a sequence of ensemble pairs, we arrange their corresponding violin plots in parallel to compare them and track the trend encoded in them.

The violin plots support two types of interactions. First, users can click on a violin plot to select an ensemble pair for further exploration. To focus on the selected ensemble pair, we enlarge the corresponding violin plot, meanwhile, compress and push other violin plots aside. When focusing on a violin plot of interest, users can explore different sub-ranges of the two PDFs via brushing. The brushed region in the current violin plot will be propagated to other violin plots to form a selection band, which reveals the distribution of the currently selected members in other ensembles. Figure 6(b) illustrates how we compute the band. The left PDF is in focus now and the brushed region selects members 1, 3, and 4. We compute the mean and standard deviation of the three members' likelihood scores in the current violin plot, as well as all other violin plots (e.g., the right violin plot in Figure 6(b)). Connecting the one standard deviation range of the mean likelihood scores across all violin plots forms a selection band. The two selections in the violin plot in Figure 5(a2) (i.e., the two brushed regions) lead to two bands. Comparing the width of the bands and the overlap between the bands provides useful insight in understanding the two selected sets of members across all ensemble pairs (details in Section 6.1).

5.1.2. Member View

The member view (Figure 5(b)) is used to visualize and compare the selected members (i.e., members in the brushed region of the violin plot). After users select members of interest in both ensembles, the selected members in each ensemble are aggregated into a mean field. The two mean fields are then visualized in two juxtaposed views as two heat maps (for 2D fields) or two volume renderings (for 3D fields) for comparison. Users can also check individual selected members by selecting them from the drop-down list shown in the header of this view (Figure 5(b), by default, the view presents the mean field). When users update their selections in the violin plot, the content of the member view is updated accordingly to synchronize the selection.

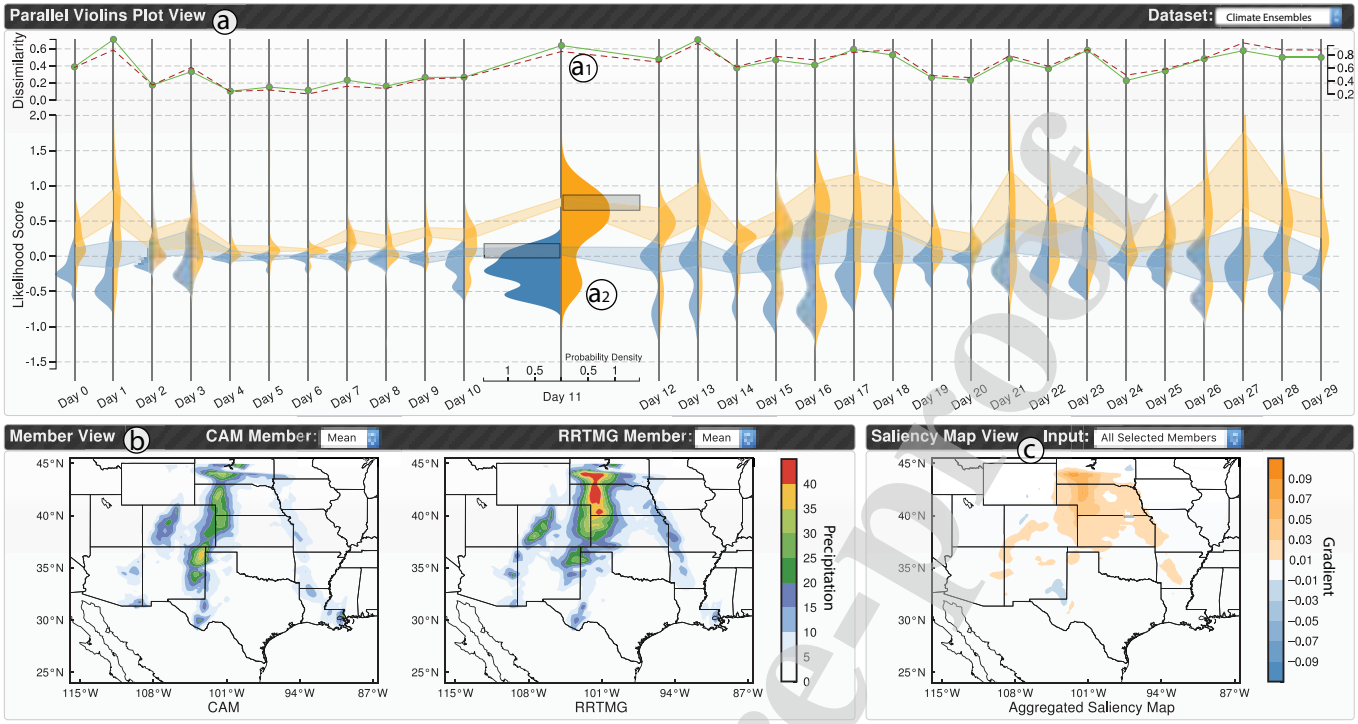


Fig. 5. User interface of CECV demonstrated by the climate ensembles generated with two different simulation models: (a) the parallel violins plot view presents the overall dissimilarities (a1) as well as the likelihood score distributions (a2) of the ensemble pairs in comparing; (b) after selecting an ensemble pair of interest and brushing on the corresponding violin plot, members whose likelihood score are within the brushed ranges are visualized in the member view; (c) by feeding the selected members into the trained discriminative network, saliency maps are generated using backprogation and visualized in the saliency map view.

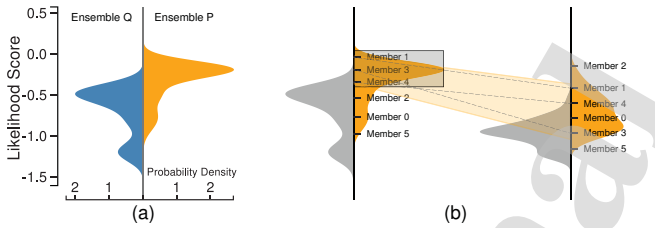


Fig. 6. (a) Jointing two PDFs as side-by-side views for comparison. (b) A band is created after brushing on a violin plot, which reveals the distribution of the currently selected members in other ensembles.

5.1.3. *Saliency Map View*

The saliency map view (Figure 5(c)) targets to demonstrate the result from our third level comparative analysis for the selected members, i.e., the spatial regions that are the most important for differentiating the two ensembles. Specifically, each selected member is fed into the trained discriminative network and the gradient of the likelihood score with respect to this member is computed using backpropagation to generate a saliency map (details in Section 4.2). The saliency maps from different selected members are then aggregated into a mean saliency map, and visualized using heat map (for 2D fields) or volume rendering (for 3D fields). To differentiate the positive gradients and negative gradients in the aggregated saliency map, a diverging color map (blue-white-orange) is used in this view. Same as the member view, the saliency map view is also linked with the violin plot view, i.e. any updates in the violin plot will trigger the updates in this view as well.

5.2. *Exploration Guidelines*

In exploring and comparing the ensembles using our visualization system, a top-down exploration strategy is recommended, which follows Shneiderman's information seeking mantra, i.e., *"Overview first, zoom and filter, then details-on-demand"* (Shneiderman, 1996) .

Overview First The PVP view (Figure 5(a)) provides the overall dissimilarities and the likelihood score distributions of the ensemble pairs in comparing. By comparing the dissimilarities and likelihood score distributions, users can obtain an overview of which pairs of ensembles are more similar/dissimilar with each other. From there, they can identify ensemble pairs of interest for further exploration.

Zoom and Filter After identifying an ensemble pair of interest (e.g. an ensemble pair with a large dissimilarity value), users can focus on this pair by clicking on the corresponding violin plot. This interaction will highlight the current violin plot by zooming into it, meanwhile, compressing and pushing other violin plots aside, as shown in Figure 5(b). From the currently focused violin plot, users can brush on either (or both) side(s) of the violin plot (representing the two ensembles) to select subsets of members for further exploration and analysis.

Details-on-Demand When users select a subset of members from each ensemble, the selected members will be visualized and compared in the member view (Figure 5(b)). In addition, the selected members are also fed into the trained discriminative network to generate their saliency maps, which will be aggregated and visualized in (Figure 5(c)). From the saliency

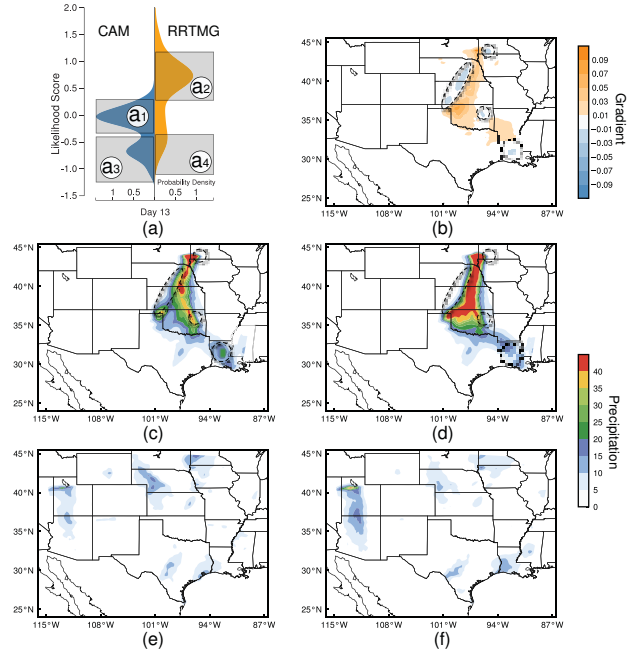


Fig. 7. Comparison between two models in day 13: (a) likelihood score distributions and selected ranges; (b) aggregated saliency map of the selected members; (c), (d), (e), and (f) mean of the selected members within the ranges (a1), (a2), (a3), and (a4), respectively.

map view, important spatial regions in differentiating the two ensembles can be located.

6. Case Studies

We demonstrate the effectiveness of CECV with two real-world applications: one compares climate ensembles generated with different simulation models (Section 6.1), the other compares CFD ensembles generated with different spatial resolutions (Section 6.2).

6.1. Climate Ensembles with Different Simulation Models

The climate ensembles were generated using the weather research and forecasting (WRF) regional climate model with two different physical sub-models: CAM vs. RRTMG. The simulation domain was located over the southern great plains region (latitude: $25°N$–$44°N$, longitude: $112°W$–$90°W$) with the grid spacing of 25 km, i.e., the spatial resolution is 87 (latitude) × 89 (longitude). The simulations generated an ensemble for each sub-model using 150 different parameter settings to predict the precipitation over the 30 days of June 2007. As a result, for each sub-model, an ensemble of 150 members were produced for each day of June 2007. Using our three-level collective comparison approach, we compare the two ensembles in our visualization system.

6.1.1. Level I: Dissimilarity Comparison

The overall dissimilarities for the ensemble pairs of the 30 days are visualized as a line chart in Figure 5(a1). We first evaluate the accuracy of the proposed method (green curve in Figure 5(a1)) with the earth mover's distances (red curve in Figure 5(a1)) estimated based on (Rubner et al., 1998). We can see that the dissimilarities estimated with the two methods have a similar trend over time. From the line chart, days that the two models highly agree (e.g. day 4, 5, and 6) or disagree (e.g. day 2, 11, and 13) with each other can be easily identified. With the guidance from the overall dissimilarities, representative days can be selected for further exploration, such as the days with a higher dissimilarity between the two models.

6.1.2. Level II: Member Comparison

From the first level comparison, day 11 attracted our attention the most, because the two ensembles in that day have a large dissimilarity value. After selecting day 11, the likelihood score distributions of the two models are highlighted in Figure 5(a2). By comparing the two likelihood score distributions, we can see that the two distributions share some common ranges with non-zero probabilities, indicating the results generated from the two models agree with each other in the corresponding members. We can also see that the distribution of the RRTMG model (i.e. the orange PDF) covers some ranges in which the CAM model (i.e. the blue PDF) has zero probabilities. These ranges indicate that the corresponding members in the RRTMG model are dramatically different from members in the CAM model. By brushing on the two distributions, users can select members whose likelihood score falls in a certain range. For example, in Figure 5(a2), two sets of members from the two ensembles are selected. With these selections, two bands (one for each set of members) are also generated. From the two bands, we found that the selected members rarely overlap with each other, which means the simulations disagree with each other in those members across time. The mean fields of the two sets of selected members in Day 11 are visualized and compared in Figure 5(b). We observed that the precipitation is higher in the Central United States than other regions in both sets of the selected members. However, the selected members from the RRTMG model (which have higher likelihood scores) have higher precipitation in the Central United States regions than the selected members from the CAM model (which have lower likelihood scores).

The ensembles for Day 13 also have very high dissimilarity (as shown in the line chart in Figure 5(a1)). To explore the members from the two ensembles, four ranges were selected by brushing on the two likelihood distributions, as shown in Figures 7(a1–a4). The mean fields of the members in the ranges indicated by Figures 7(a1) and 7(a2) are shown in Figures 7(c) and 7(d), respectively. Similar to day 11, we found that the two models predict high precipitation in the Central United States, and the precipitation predicted by the RRTMG model is higher than that of the CAM model in general. Figures 7(e) and 7(f) show the mean fields of the members in the ranges indicated by 7(a3) and 7(a4). While the simulations agree with each other on some members in those ranges, the CAM model has more members with likelihood scores within the ranges indicated by 7(a3) and 7(a4).

By exploring and comparing the members of the two simulation models, we found that: (1) in general the two simulation models predict high precipitation in similar spatial regions; (2) the RRTMG model tends to produce members with higher pre-

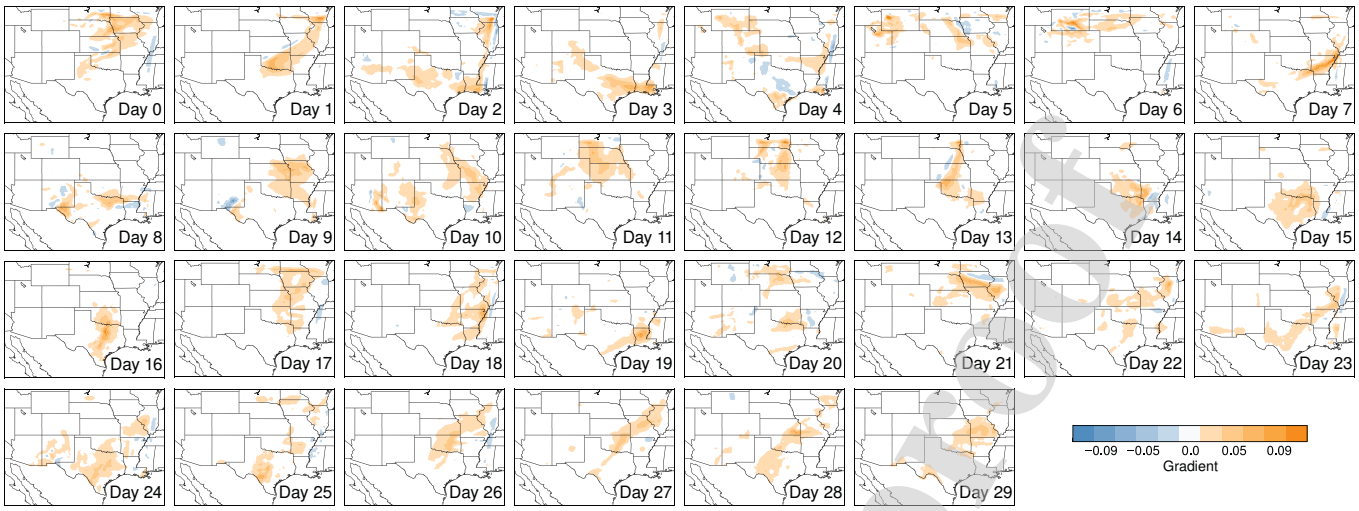


Fig. 8. Aggregated saliency map of all members for each day. Positive gradients cover larger spatial regions than negative gradients.

cipitation than the CAM model in those regions.

6.1.3. Level III: Region Comparison

Focusing on the selected members of the two ensembles in day 13 (Figure 7), we drill down to the third level comparative analysis to further investigate what spatial regions are the most important for differentiating the two ensembles. The mean of the saliency maps for the selected members is visualized in 7(b). We found that the sensitive spatial regions (i.e. regions with high gradient magnitude) are located at regions with high precipitation in both models. We can also see that there are two large spatial regions that have positive gradients, which means if the precipitation in these regions increases, the corresponding members are more likely to be generated by the RRTMG model. There are also several small regions with negative gradients (marked by circles in Figure 7), which indicates that increasing precipitation in these regions will decrease the likelihood score (i.e. make the member more likely to be generated by the CAM model).

We also performed similar explorations on the other days to fully understand the two ensembles. Figure 8 shows the aggregated saliency maps of all members across the 30 days. We can find that in most of the spatial regions, the gradients are zero (i.e. white space in Figure 8). In these spatial regions, the value of the precipitation is not affecting the likelihood score of the members. Positive gradients cover larger spatial regions than negative gradients across all 30 days. Hence, compared with the CAM model, the RRTMG model has higher probability to produce higher precipitation in these regions.

6.2. CFD Ensembles with Different Spatial Resolutions

The CDF particle ensembles we studied in this section is from the 2016 IEEE SciVis Contest[1], which were generated with ensemble simulations using the finite pointset method. The simulations modeled the diffusion of salt from the top surface of a cylindrical flow domain that is filled with pure water. The simulation domain was represented by a cloud of discrete particles, each of which was endowed with the relevant local properties of the data, such as velocity and salt concentration. Three ensembles with different spatial resolutions (i.e. different numbers of particles) were generated, and we use E_l (around $250,000$ particles), E_m (around $650,000$ particles), and E_h (around $1,900,000$ particles) to index them. These three ensembles have 48, 23, and 22 members, respectively. Each member is a sequence of particle clouds, which record the temporal evolution of particles. Because adaptive time stepping is used in the simulation, differed members may record the state of the particles at different sets of timesteps.

The goal of this simulation is to compare the salt concentration at different spatial locations over time. In this application, we first converted the particles in individual member from all three ensembles into a density field of $64 \times 64 \times 64$ grid points using density estimation. Second, we resampled the resulting density fields in the temporal domain using linear interpolation to make the ensembles share the same timesteps. There are 40 timesteps in total after the resampling. Finally, we applied our collective comparison approach to compare ensembles at individual timesteps, and analyzed the results from all 40 timesteps.

6.2.1. Level I: Dissimilarity Comparison

Figure 9(a) shows the overall dissimilarities for the three ensemble pairs of the 40 timesteps. We found that the dissimilarities between E_l and E_h is the largest, and the dissimilarities between E_m and E_h is the smallest across all timesteps. We also found that the dissimilarity increases over time for all three ensemble pairs. For further exploration we focused on the comparison between E_l and E_h.

6.2.2. Level II: Member Comparison

Figure 9(b) shows the likelihood score distributions of E_l and E_h. We can see that across all timesteps, none of the two distributions have overlapped ranges, which means the two ensembles do not share any similar members. We can also see that

[1] http://www.uni-kl.de/sciviscontest/

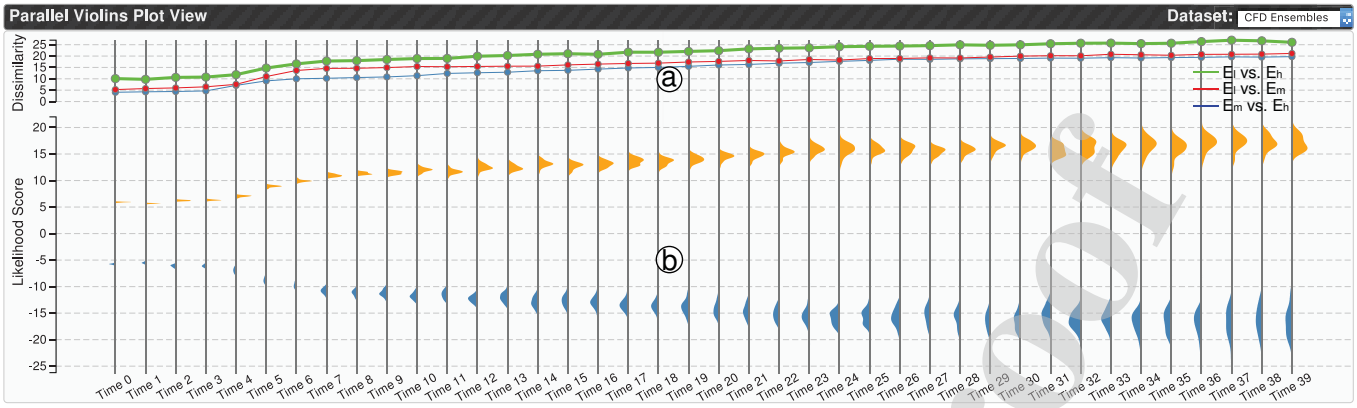


Fig. 9. The PVP for the CFD ensembles: (a) overall dissimilarities across three resolutions; (b) likelihood score distributions of E_l and E_h.

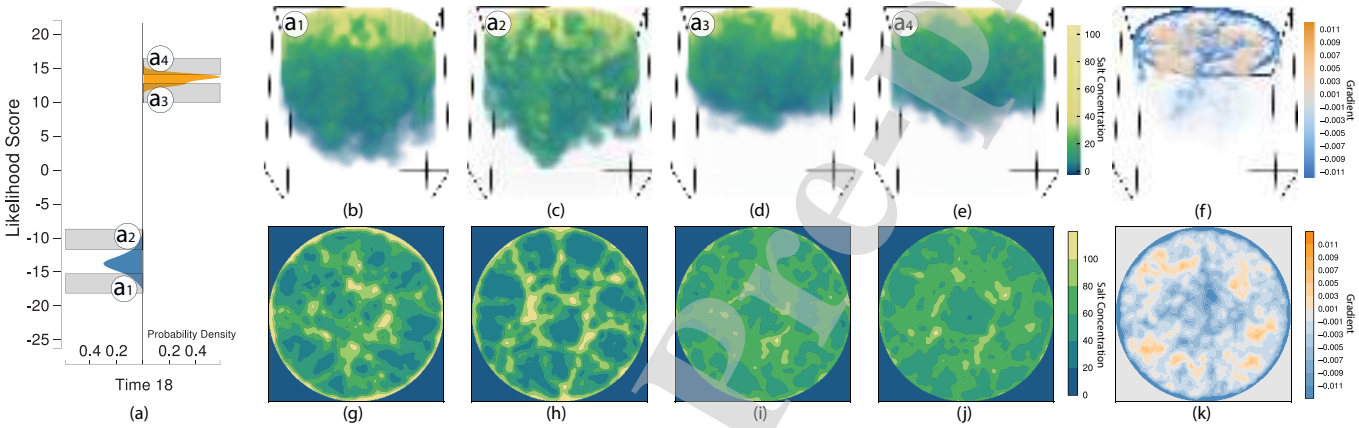


Fig. 10. Visualization of the selected members in time 18: (a) the likelihood distributions of E_l and E_h as well as the selected ranges; (b)–(e) mean fields of the selected members within the ranges shown in (a1)–(a4), respectively; (g)–(j) slice on the top of the domain for the four mean fields; (f) and (k) mean saliency map of selected members and the slice of the saliency map.

distance between the likelihood score distributions of the two ensembles increases over time.

To analyze how the members of the two ensembles are different, we selected timestep 18 for further exploration. Figure 10 shows the violin plot of two distributions at timestep 18, and 4 ranges are selected as shown in Figures 10(a1–a4). We compared E_l and E_h by visualizing and analyzing the members of the four selected ranges. The mean of the members within the ranges indicated by Figures 10(a1–a4) are shown in Figures 10(b–e), respectively. We observed that the members of E_l have higher concentration at the lower half region of the domain than the members of E_h. By further comparing a slice on the top of the domain for the four mean fields in Figures 10(g–j), we can see that the members of E_l typically have higher concentration at the boundary of the cylindrical domain and a few small regions in near the center of the domain. We can also see that the members of E_l have large number of regions with concentration smaller than 20. Compared with E_l, the members of E_h do not have regions with extreme high or low concentration in the slice, which means the concentration are nearly evenly distributed in the slice. Compared with Figures 10(g) and (h), we found that the members with higher likelihood scores (i.e. members within the range of Figure 7(a2)) have lower concentration in the boundary than the members with lower likelihood scores (i.e. members within the range of Figure 7(a1)). Hence, members with higher concentration in the boundary are considered more likely to be sampled from E_l.

6.2.3. Level III: Region Comparison

We further visualized and analyzed the sensitivity of spatial regions for differentiating the two ensembles based on the third level comparative analysis. Figure 10(f) shows the mean saliency map of the selected members, and Figure 10(k) shows a slice on the top of the domain of the mean saliency map. Regions with low gradient magnitude are rendered with low transparency in the volume visualization. We found that regions with high gradient magnitude are located at the top of the domain, which indicates that in those regions the value of the salt concentration is important for differentiating the two ensembles.

High negative gradients are concentrated at the boundary of the cylindrical domain on the top, which means if the salt concentration at these regions increases, the member is more likely to come from E_l. Negative gradients can also be found in the lower half of the domain, which is because the members of E_l have higher concentration at the lower half of the domain than the members of E_h. We also observed that on the top of the domain, there are several regions with positive gradient. This is because the members of E_l have regions with extreme low

Table 1. Architecture details and training time of the discriminative networks used in the two case studies.

Dataset	Convolutional Layers				Fully Connected Layers		# Pairs	Batch Size	# Iterations	# Nodes	Time (mins)
	# Layers	# Filters	Filter Size	Strides	# Layers	# Neurons					
Climate Ensembles	3	[8, 16, 32]	5×5	2×2	1	4224	30	50	4000	30	1.92
CFD Ensembles	3	[16, 32, 64]	$3\times3\times3$	$2\times2\times2$	1	32768	120	20	4000	40	17.95

concentration on the top, and the concentration of members of E_h at those regions are higher.

7. Implementation and Performance

The proposed approach consists of two modules: the training of the discriminative networks, and the visualization system to support the collective comparison of ensemble pairs based on the trained discriminative networks. The training of discriminative networks is implemented using TensorFlow[2]. The architecture details of the discriminative networks used in the two applications are listed in Table 1. The visualization system is implemented based on a web server/client framework. The PVP view is implemented using D3.js on the client side. The members and the saliency maps are rendered using OpenGL on the server side and sent to the client to visualize.

The most compute-intensive part of our method is the training of the discriminative networks. Because the training of the discriminative network for each ensemble pair is independent from other discriminative networks, we parallelized the training of the discriminative networks over the ensemble pairs. In our implementation, the ensemble pairs are assigned to multiple processes, and each process trains a subset of the ensemble pairs. We tested the performance of the training on Owens at the Ohio Supercomputer Center, which contains 160 nodes. Each node has an Intel Xeon E5-2680 CPU, an NVIDIA Pascal P100 GPU, 128 GB main memory, and 16 GB GPU memory. Our benchmark used up to 40 nodes with GPUs, and the performances of the two applications are listed in Table 1. From our observation, in most cases, the loss converged after 2000 iterations. In the experiments, the maximum number of iterations was set to 4000 to ensure the convergence of loss for all cases.

8. Discussion and Future Work

In this section, we first discuss our choice of hyperparameters (i.e. parameters whose value are fixed during training) for the discriminative networks. Then other divergences between distributions and their corresponding objective functions are discussed. Then we discuss the domain expert's feedback, limitations of our approach, and potential future work to address them.

Hyperparameters of Discriminative Networks The most important hyperparameters of a discriminative network include number of layers, number of neurons per layer, and the size of the convolutional filters. Few routine approaches or practical recommendations have been reported to guide the selection of those hyperparameters, and the values of them are often data dependent. In this work, we resort to the following heuristic, which is commonly adopted by deep learning experts, to select the value for those hyperparameters. First, we start with an estimated size of the neural network, e.g., 3 convolutional layers, 32 filters per layer, and the size of each filter is 5×5. Second, by observing the training losses, we gradually increase/decrease those values, so that the network could accomplish our training goals with a rather smaller size (to reduce the training time). The final values of those hyperparameters used in this work are reported in Table 1. They may not be the optimal selections, but the successful training results reported in our paper have proved that they are proper choices, and deriving the optimal hyperparameters is out of the scope of this paper.

Divergences and Objective Functions While we focus on the Wasserstein distance in this work, the proposed approach is flexible to estimate other widely used divergences such as the Kullback-Leibler divergence. The divergences are corresponding to different objective functions, which can be written in a general form as:

$$L(P,Q;\phi) = \mathbb{E}_{p\sim P}[\,\mu(\mathcal{D}_\phi(p))\,] - \mathbb{E}_{q\sim Q}[\,\upsilon(\mathcal{D}_\phi(q))\,], \quad (4)$$

where μ and υ are two specific functions that control the type of the divergence related to the objective function. Table 2 shows various divergences between distributions and their corresponding functions μ and υ. More divergences and detailed mathematical proof can be found in (Sriperumbudur et al., 2010; Nguyen et al., 2010; Nowozin et al., 2016).

Table 2. Defined μ and υ functions for different types of divergences between distributions. $\mathcal{F}$ is a class of real-value bounded functions, and $\|\mathcal{D}_\phi\|_1$ stands for 1-Lipschitz functions.

Divergence	μ	υ	Function Class of $\mathcal{D}_\phi$
Kullback-Leibler	$\mathcal{D}_\phi$	$e^{\mathcal{D}_\phi}$	$\mathcal{F}$
Jensen-Shannon	$\log\frac{2}{1+e^{-\mathcal{D}_\phi}}$	$-\log\frac{2}{1+e^{\mathcal{D}_\phi}}$	$\mathcal{F}$
Pearson χ^2	$\mathcal{D}_\phi$	$\frac{1}{4}\mathcal{D}_\phi^2+\mathcal{D}_\phi$	$\mathcal{F}$
Squared Hellinger	$1-e^{\mathcal{D}_\phi}$	$e^{-\mathcal{D}_\phi}-1$	$\mathcal{F}$
Total Variation	$\frac{1}{2}\tanh(\mathcal{D}_\phi)$	$\frac{1}{2}\tanh(\mathcal{D}_\phi)$	$\mathcal{F}$
Wasserstein	$\mathcal{D}_\phi$	$\mathcal{D}_\phi$	$\mathcal{F}\colon \|\mathcal{D}_\phi\|_1$

Domain Expert Feedback and Future Work We verified the effectiveness and usefulness of the proposed approach with one domain expert, who is from environmental sciences and working with weather models. Overall, the expert commented that our approach provided a good guidance to explore and compare multiple ensembles at different levels, and the visualization interface is intuitive and friendly to him. Specifically, the expert commented that reducing members into simple representations (i.e. likelihood scores) is helpful for him to visualize and analyze the differences between complicated ensembles, and the saliency maps are useful to identify important spatial regions. He also mentioned that our approach is an

[2]https://www.tensorflow.org/

important step toward revealing the complex input/output relations driven by dynamical and physical sensitivities in weather and climate models. Meanwhile, the expert also mentioned several limitations of the proposed method. First, the proposed approach compares ensembles at different time steps independently, which does not consider the temporal coherence of the data. Hence, in the future, we would like to extend our method to study the spatial and temporal coherence of time-varying data with deep sequential models. Second, the generalization of the proposed method to unseen samples depends heavily on the quality of the existing samples. If the samples are not able to represent the underlying distribution accurately, the proposed method is not able to be generalized to other samples from the distribution. In the future, we would like to explore methods that are less sensitive to the quality of the samples.

9. Conclusion

In this paper, we present a method to collectively compare and visualize ensembles using deep discriminative neural networks. Our method compares a pair of ensembles by training a discriminative neural network, which takes the two ensembles as input. The output from the network enables to compare the pair of ensembles from three different levels (in a top-down order): overall dissimilarity level; member level; and spatial region level. We further extend our method to the comparison of multiple ensemble pairs to copy with real-world ensemble data. To present our comparison results and facilitate the visual analytics of complex ensembles, we design and develop a visualization system. The system demonstrates our three-level comparative analysis results via multiple coordinated views (i.e., line chart, violin plots, and multiple juxtaposed spatial views). Through case studies on real-world ensembles from different disciplines, we validate the effectiveness and usefulness of our approach with domain scientists.

Acknowledgments

This work was supported in part by US Department of Energy Los Alamos National Laboratory contract 47145 and UT-Battelle LLC contract 4000159447 program manager Laura Biven.

References

Alabi, O.S., Wu, X., Harter, J.M., Phadke, M., Pinto, L., Petersen, H., Bass, S., Keifer, M., Zhong, S., Healey, C., Taylor II, R.M., 2012. Comparative visualization of ensembles using ensemble surface slicing, in: Prooceedings of SPIE, pp. 8924:1–12.

Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein GAN. arXiv preprint arXiv:1701.07875 .

Athawale, T., Entezari, A., 2013. Uncertainty quantification in linear interpolation for isosurface extraction. IEEE Transactions on Visualization and Computer Graphics 19, 2723–2732.

Athawale, T., Sakhaee, E., Entezari, A., 2016. Isosurface visualization of data with nonparametric models for uncertainty. IEEE Transactions on Visualization and Computer Graphics 22, 777–786.

Bensema, K., Gosink, L., Obermaier, H., Joy, K.I., 2016. Modality-driven classification and visualization of ensemble variance. IEEE Transactions on Visualization and Computer Graphics 22, 2289–2299.

Biswas, A., Lin, G., Liu, X., Shen, H.W., 2017. Visualization of time-varying weather ensembles across multiple resolutions. IEEE Transactions on Visualization and Computer Graphics 23, 841–850.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010, pp. 177–186.

Chen, H., Zhang, S., Chen, W., Mei, H., Zhang, J., Mercer, A., Liang, R., Qu, H., 2015. Uncertainty-aware multidimensional ensemble data visualization and exploration. IEEE Transactions on Visualization and Computer Graphics 21, 1072–1086.

Chwialkowski, K.P., Ramdas, A., Sejdinovic, D., Gretton, A., 2015. Fast two-sample testing with analytic representations of probability measures, in: Proceedings of NIPS, pp. 1981–1989.

Demir, I., Kehrer, J., Westermann, R., 2016. Screen-space silhouettes for visualizing ensembles of 3D isosurfaces, in: Proceedings of 2016 IEEE Pacific Visualization Symposium, pp. 204–208.

Diggle, P., Heagerty, P., Liang, K.Y., Zeger, S., 2013. Analysis of Longitudinal Data. Oxford University Press, Oxford.

Dutta, S., Chen, C.M., Heinlein, G., Shen, H.W., Chen, J.P., 2017a. In situ distribution guided analysis and visualization of transonic jet engine simulations. IEEE Transactions on Visualization and Computer Graphics 23, 811–820.

Dutta, S., Shen, H.W., 2016. Distribution driven extraction and tracking of features for time-varying data analysis. IEEE Transactions on Visualization and Computer Graphics 22, 837–846.

Dutta, S., Woodring, J., Shen, H.W., Chen, J.P., Ahrens, J., 2017b. Homogeneity guided probabilistic data summaries for analysis and visualization of large-scale data sets, in: Proceedings of 2017 IEEE Pacific Visualization Symposium, pp. 111–120.

Ferstl, F., Bürger, K., Westermann, R., 2016a. Streamline variability plots for characterizing the uncertainty in vector field ensembles. IEEE Transactions on Visualization and Computer Graphics 22, 767–776.

Ferstl, F., Kanzler, M., Rautenhaus, M., Westermann, R., 2016b. Visual analysis of spatial variability and global correlations in ensembles of iso-contours. Computer Graphics Forum 35, 221–230.

Ferstl, F., Kanzler, M., Rautenhaus, M., Westermann, R., 2017. Time-hierarchical clustering and visualization of weather forecast ensembles. IEEE Transactions on Visualization and Computer Graphics 23, 831–840.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Proceedings of NIPS, pp. 2672–2680.

Gosink, L., Bensema, K., Pulsipher, T., Obermaier, H., Henry, M., Childs, H., Joy, K.I., 2013. Characterizing and visualizing predictive uncertainty in numerical ensembles through bayesian model averaging. IEEE Transactions on Visualization and Computer Graphics 19, 2703–2712.

Gretton, A., Borgwardt, K.M., Rasch, M., Schölkopf, B., Smola, A.J., 2007. A kernel method for the two-sample-problem, in: Proceedings of NIPS, pp. 513–520.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C., 2017. Improved training of wasserstein gans, in: Proceedings of NIPS, pp. 5767–5777.

Guo, H., He, W., Peterka, T., Shen, H.W., Collis, S.M., Helmus, J.J., 2016. Finite-time Lyapunov exponents and Lagrangian coherent structures in uncertain unsteady flows. IEEE Transactions on Visualization and Computer Graphics 22, 1672–1682.

Hazarika, S., Biswas, A., Shen, H.W., 2018. Uncertainty visualization using copula-based analysis in mixed distribution models. IEEE Transactions on Visualization and Computer Graphics 24, 934–943.

He, W., Liu, X., Shen, H.W., Collis, S.M., Helmus, J.J., 2017. Range likelihood tree: A compact and effective representation for visual exploration of uncertain data sets, in: Proceedings of 2017 IEEE Pacific Visualization Symposium, pp. 151–160.

Hintze, J.L., Nelson, R.D., 1998. Violin plots: A box plot-density trace synergism. The American Statistician 52, 181–184.

Hlawatsch, M., Leube, P., Nowak, W., Weiskopf, D., 2011. Flow radar glyphs-static visualization of unsteady flow with uncertainty. IEEE Transactions on Visualization and Computer Graphics 17, 1949–1958.

Höllt, T., Magdy, A., Zhan, P., Chen, G., Gopalakrishnan, G., Hoteit, I., Hansen, C.D., Hadwiger, M., 2014. Ovis: A framework for visual analysisof ocean forecast ensembles. IEEE Transactions on Visualization and Computer Graphics 20, 1114–1126.

Hummel, M., Obermaier, H., Garth, C., Joy, K.I., 2013. Comparative visual analysis of Lagrangian transport in CFD ensembles. IEEE Transactions on

Visualization and Computer Graphics 19, 2743–2752.

Im, D.J., Ma, H., Taylor, G., Branson, K., 2018. Quantitatively evaluating gans with divergences proposed for training, in: Proceedings of International Conference on Learning Representations.

Jarema, M., Demir, I., Kehrer, J., Westermann, R., 2015. Comparative visual analysis of vector field ensembles, in: Proceedings of 2015 IEEE Conference on Visual Analytics Science and Technology (VAST), pp. 81–88.

Jitkrittum, W., Szabó, Z., Chwialkowski, K.P., Gretton, A., 2016. Interpretable distribution features with maximum testing power, in: Proceedings of NIPS, pp. 181–189.

Kehrer, J., Muigg, P., Doleisch, H., Hauser, H., 2011. Interactive visual analysis of heterogeneous scientific data across an interface. IEEE Transactions on Visualization and Computer Graphics 17, 934–946.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Köthur, P., Witt, C., Sips, M., Marwan, N., Schinkel, S., Dransch, D., 2015. Visual analytics for correlation-based comparison of time series ensembles. Computer Graphics Forum 34, 411–420.

Kumpf, A., Tost, B., Baumgart, M., Riemer, M., Westermann, R., Rautenhaus, M., 2018. Visualizing confidence in cluster-based ensemble weather forecast analyses. IEEE Transactions on Visualization and Computer Graphics 24, 109–119.

Lopez-Paz, D., Oquab, M., 2016. Revisiting classifier two-sample tests. arXiv preprint arXiv:1610.06545 .

Lu, K., Chaudhuri, A., Lee, T.Y., Shen, H.W., Wong, P.C., 2013. Exploring vector fields with distribution-based streamline analysis, in: Proceedings of 2013 IEEE Pacific Visualization Symposium, pp. 257–264.

Mirzargar, M., Whitaker, R.T., Kirby, R.M., 2014. Curve boxplot: Generalization of boxplot for ensembles of curves. IEEE Transactions on Visualization and Computer Graphics 20, 2654–2663.

Moon, K.R., Hero, A.O., 2014. Multivariate f-divergence estimation with confidence. arXiv preprint arXiv:1411.2045 .

Müller, A., 1997. Integral probability metrics and their generating classes of functions. Advances in Applied Probability 29, 429–443.

Nguyen, X., Wainwright, M.J., Jordan, M.I., 2010. Estimating divergence functionals and the likelihood ratio by convex risk minimization. IEEE Transactions on Information Theory 56, 5847–5861.

Nowozin, S., Cseke, B., Tomioka, R., 2016. f-GAN: Training generative neural samplers using variational divergence minimization, in: Proceedings of NIPS, pp. 271–279.

Obermaier, H., Joy, K.I., 2014. Future challenges for ensemble visualization. IEEE Computer Graphics and Applications 34, 8–11.

Pfaffelmoser, T., Westermann, R., 2013. Visualizing contour distributions in 2D ensemble data, in: Proceedings of EuroVis-Short Papers, pp. 55–59.

Phadke, M.N., Pinto, L., Alabi, O., Harter, J., Taylor II, R.M., Wu, X., Petersen, H., Bass, S.A., Healey, C.G., 2012. Exploring ensemble visualization. Proc. SPIE 8294, 1–12.

Poczos, B., Schneider, J., 2011. On the estimation of α-divergences, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 609–617.

Póczos, B., Xiong, L., Schneider, J.G., 2012. Nonparametric divergence estimation with applications to machine learning on distributions. arXiv preprint arXiv:1202.3758 .

Pöthkow, K., Hege, H.C., 2011. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. IEEE Transactions on Visualization and Computer Graphics 17, 1393–1406.

Pöthkow, K., Hege, H.C., 2013. Nonparametric models for uncertainty visualization. Computer Graphics Forum 32, 131–140.

Pöthkow, K., Weber, B., Hege, H.C., 2011. Probabilistic marching cubes. Computer Graphics Forum 30, 931–940.

Potter, K., Wilson, A., Bremer, P.T., Williams, D., Doutriaux, C., Pascucci, V., Johnson, C.R., 2009. Ensemble-vis: A framework for the statistical visualization of ensemble data, in: Proceedings of 2009 IEEE International Conference on Data Mining Workshops, pp. 233–240.

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 .

Raj, M., Mirzargar, M., Preston, J.S., Kirby, R.M., Whitaker, R.T., 2016. Evaluating shape alignment via ensemble visualization. IEEE Computer Graphics and Applications 36, 60–71.

Rubner, Y., Tomasi, C., Guibas, L.J., 1998. A metric for distributions with applications to image databases, in: Proceedings of Sixth International Conference on Computer Vision, pp. 59–66.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representations by error propagation, in: Proceedings of Parallel Distributed Processing, pp. 318–362.

Sakhaee, E., Entezari, A., 2017. A statistical direct volume rendering framework for visualization of uncertain data. IEEE Transactions on Visualization and Computer Graphics 23, 2509–2520.

Sanyal, J., Zhang, S., Dyer, J., Mercer, A., Amburn, P., Moorhead, R.J., 2010. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. IEEE Transactions on Visualization and Computer Graphics 16, 1421–1430.

Shneiderman, B., 1996. The eyes have it: A task by data type taxonomy for information visualizations, in: Proceedings of the 1996 IEEE Symposium on Visual Languages, pp. 336–343.

Simonyan, K., Vedaldi, A., Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 .

Sriperumbudur, B.K., Fukumizu, K., Gretton, A., Schölkopf, B., Lanckriet, G.R.G., 2010. Non-parametric estimation of integral probability metrics, in: Proceedings of 2010 IEEE International Symposium on Information Theory, pp. 1428–1432.

Wang, J., Liu, X., Shen, H.W., Lin, G., 2017. Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. IEEE Transactions on Visualization and Computer Graphics 23, 81–90.

Wang, Q., Kulkarni, S.R., Verdu, S., 2009. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. IEEE Transactions on Information Theory 55, 2392–2405.

Wei, T.H., Chen, C.M., Biswas, A., 2015. Efficient local histogram searching via bitmap indexing. Computer Graphics Forum 34, 81–90.

Wei, T.H., Chen, C.M., Woodring, J., Zhang, H., Shen, H.W., 2017. Efficient distribution-based feature search in multi-field datasets, in: Proceedings of 2017 IEEE Pacific Visualization Symposium, pp. 121–130.

Whitaker, R.T., Mirzargar, M., Kirby, R.M., 2013. Contour boxplots: A method for characterizing uncertainty in feature sets from simulation ensembles. IEEE Transactions on Visualization and Computer Graphics 19, 2713–2722.

Yang, B., Qian, Y., Lin, G., Leung, R., Zhang, Y., 2012. Some issues in uncertainty quantification and parameter tuning: A case study of convective parameterization scheme in the WRF regional climate model. Atmospheric Chemistry and Physics 12, 2409–2427.