Geometry-Sensitive Ensemble Mean based on Wasserstein Barycenters: Proof-of-Concept on Cloud Simulations

Jia Li[*] Fuqing Zhang[†]

Abstract

An ensemble of forecasts generated by different model simulations provides rich information for meteorologists about impending weather such as precipitating clouds. One major form of forecasts presents cloud images created by multiple ensemble members. Common features identified from these images are often used as the consensus prediction of the entire ensemble, while the variation among the images indicates forecast uncertainty. However, the large number of images and the possibly tremendous extent of dissimilarity between them pose cognitive challenges for decision making. In this paper, we develop novel methods for summarizing an ensemble of forecasts represented by cloud images and call them collectively the *Geometry-Sensitive Ensemble Mean* (GEM) toolkit. Conventional pixel-wise or feature-based averaging either loses interesting geometry information or focuses narrowly on some pre-chosen characteristics of the clouds to be forecasted. In GEM, we represent a cloud simulation by a Gaussian mixture model, which captures cloud shapes effectively without making special assumptions. Furthermore, using a state-of-the-art optimization algorithm, we compute the Wasserstein barycenter for a set of distributional entities, which can be considered as the consensus mean or centroid under the Wasserstein metric. Experimental results on two sets of ensemble simulated images are provided. Supplemental materials for the article are available online.

Keywords: D2-clustering, Ensemble aggregation, Ensemble synthesis, Modal clustering

[*] Jia Li, corresponding author, is Professor of Statistics at the Pennsylvania State University, University Park, PA 16802. Email: jiali@psu.edu

[†] Fuqing Zhang is Professor of Meteorology and Atmosphere Science and Director of Center for Advanced Data Assimilation and Predictability Techniques at the Pennsylvania State University, University Park, PA 16802. Email: fzhang@psu.edu

1 Introduction

One important challenge for meteorologists is to effectively integrate simulated forecasting results from multiple numerical weather prediction models or many different forecast realizations by the same model. The collection of these model forecasts is called an ensemble while each forecast realization is called an ensemble member. The report by National Research Council et al. (2006) and the seminal book by Kalnay (2002) have detailed discussion on the necessity of ensembles and the demand for more effective use of ensembles in decision making. Extracting overall patterns by direct inspection of a large number of ensemble simulations is prone to subjectivity, not to mention the difficulty on the inspector (i.e., the weather forecaster). In extreme weather conditions, these decisions bear enormous economical and life-saving consequences. However, human cognition is severely limited when supplied with even a moderate number of possibilities, especially when these possible solutions (ensemble realizations) are not similar. This constitutes a classical example of "information overload" for weather forecasters. It is thus valuable if computer software can automatically summarize the ensemble simulations, in particular in search for the most likely forecast scenario(s) and the associated uncertainties.

In the inspiring work of Sivillo, Ahlquist, and Toth (1997), high-level principles are discussed for ensemble forecasting. Specifically, five fundamental problems are raised, among which is how to display information effectively from an ensemble. Despite the fact that information overload is posed as a central challenge by those authors, existing work to address the challenge is rather limited. Although a few practices have been adopted, we have yet to see systematic development of tools aimed squarely at optimal presentation of ensemble forecasts and capable of handling various types of forecasts. Our work here is an effort in this direction. Currently, examples of ensemble visualization include (1) "spaghetti diagrams" which show plan view map images containing contour lines for the quantities (e.g., clouds) from each ensemble member, and (2) "thumbnail sketches" or "postage stamps" which are a collection of individual miniature images generated from each ensemble member.

For an ensemble of simulated cloud images predicted by different members, it is challenging to synthesize them into one or a few most likely scenarios in a meaningful way although the task may seem deceivingly easy. At present, the most common practice of aggregating meteorological ensemble forecasts is to use the equally weighted average of all ensemble members as the ensemble mean and to use the standard deviation as the uncertainty estimate (Leith, 1974; Molteni et al., 1996; Toth and

Kalnay, 1997; Sivillo, Ahlquist, and Toth, 1997). A statistical approach to ensemble post-processing was pioneered by Raftery et al. (2005), who proposed Bayesian Model Averaging (BMA) to generate a weighted ensemble average. The weights depend on the prior performance of individual members of the ensemble. BMA has been found to improve forecasting performance over the equally weighted mean, and in the meanwhile the analysis provides a theoretically solid framework for quantifying both between-model and within-model uncertainty in forecasting.

For the ensemble of cloud simulations, however, because of the large disparity among members within the ensemble, direct pixel-wise averaging will result in a "smeared out" cloud image (see Figure 4 (a)), losing critical characteristics about shape, orientation, etc. To preserve the cloud property without smearing or over-smoothing as in the simple average, the ensemble member that has the highest weight or probability is sometimes used as the consensus estimate which may not be representative of the full ensemble (Melhauser and Zhang, 2012). Various clustering analyses have also been proposed to synthesize the ensemble forecasts but so far the approach has been applied to rather low-dimensional summarized forecast products such as hurricane tracks (Don et al., 2016). We call such an approach parametric. Although the low-dimensional characterization of forecasts is useful when we focus on a particular aspect such as hurricane tracks, a significant portion of the information about the original imagery data is lost. If another aspect of the forecasts is to be examined, the analysis system has to be re-designed. It is also difficult to correlate multiple features about the forecasts since the analysis is not performed in the image space. For human cognition, it is more natural to inspect synthesized images rather than synthesized summary features.

In this paper, we propose a new paradigm of creating synthesized mean (or average, centroid) images for ensembles of cloud simulations. We call the collection of methods developed here the *Geometry-Sensitive Ensemble Mean* (GEM) toolkit. We aim at addressing the limitations of existing approaches of pixel-wise averaging or low-dimensional feature-based summary. We have developed three schemes for synthesizing images. These schemes are under the same principles of synthesis, sharing fundamentals such as the image representation and the mathematical formulation for aggregating the representations. Their differences are motivated by the various end users' preferences for the final appearance of the synthesized image. Although we use Bayesian posterior mean, which is similar to Raftery et al. (2005), our purpose and setup are different. We focus on effective visualization of an ensemble under the scenario that prior information on the performance of ensemble members is not available. The Bayesian posteriors are determined based on distances to a centroid representation of the images. Our work also differs from Melhauser and Zhang (2012), which assumes that assessment of the likelihood of the members is given by additional information. The Bayesian averaging method follows in spirit the empirical Bayesian idea. The centroid of the ensemble helps to set the prior on the ensemble members. When finding a representative member from the ensemble, our criterion is to be closest to the centroid, a stark contrast from that of Melhauser and Zhang (2012).

Our main contribution is the development of methods to summarize cloud simulation images in a geometrically meaningful way. A major novelty of our work is the GEM framework itself, which differs profoundly from current practices. In existing work, images are treated as vectors of pixel intensities. We propose a two-tier signature, a novel object-level representation, so that geometric traits of cloud patches can be retained. Although we use an existing algorithm of Ye et al. (2017) for efficient computation of the Wasserstein barycenter, the solution to barycenter alone does not trivially yield the summary images. In fact, the barycenter only determines the center locations of the cloud patches. To generate each cloud patch in the aggregated mean image, we have invented a method for integrating the shape and intensity information of the cloud patches in the ensemble member images.

The rest of the paper is organized as follows. In Section 2, we explain the rationale for the framework of GEM and provide preliminaries on Wasserstein distance and Wasserstein barycenter. We describe the algorithms for the components in GEM and the evaluation methods in Section 3. Experimental results are provided in Section 4. Finally, we conclude and discuss future work in Section 5.

2 Preliminaries

For meteorologists, the cloud intensity at one pixel is too localized to be of practical interest. On the other hand, a completely parametric approach will impose strong assumptions on the shapes of the clouds, which may often deviate grossly from reality, hence limiting applicability. The core issue here is thus to achieve simultaneously *sensitivity* to the geometric characteristics of clouds and *adaptivity* to diverse kinds of clouds simulated by different ensemble members. In this study, we propose two novel strategies in synthesizing the ensemble simulations: (1) the definition of an effective cloud-map representation (also called *signature*) with an unconventional mathematical structure and (2) the implementation of new machine learning tools for clustering and summarizing the signatures. Specifically, we want to address the following two key questions:

- *Q1: How can we formulate a cloud-map representation that captures explicitly and accurately the essential geometric characteristics of clouds and in the mean time is flexible enough for treating diverse cloud shapes?*

- *Q2: How can we efficiently create a consensus cloud simulation, that is, "ensemble centroid", such that key geometric characteristics of clouds in individual simulations are retained?*

To address the first question (*Q1*), we propose a *two-tier signature* to represent a cloud map. The image is first segmented into a set of *cloud patches*. The mean locations of the pixels in each patch associated with the total cloud intensity values within each patch form the first tier of the representation (called *first-tier signature*). At the second tier, the shape of each cloud patch is characterized by the parameters of a Gaussian distribution fitted on the weighted pixel coordinates in the patch. The weight of each pixel is proportional to its cloud intensity. Because each cloud patch examined is relatively local, fitting one Gaussian distribution is adequate.

We call the data model of the two-tier signature a *distributional entity*, or in short a *set representation*. We use the Wasserstein metric for the discrete distributions. Wasserstein metric (Rachev, 1985) takes into consideration the underlying distances between the support points and hence has nicer geometric property than L_p norm, a point to be elaborated in Section 2.2. Theoretically speaking, we can also apply the Wasserstein metric to the set of all cloud pixels, but this is not the usual practice adopted for the pixel representation because of computational intensity. In fact, if the Wasserstein distance is to be employed, it is unnecessary to use such a high granularity, as we will show in the experiments later.

We now face the second question (*Q2*) because under the Wasserstein distance we lose the conventional notion of a mean. The usual operation of computing mean in each dimension becomes meaningless because the data are not vectors but sets containing unordered and dynamic points. Our solution is to use the Wasserstein barycenter. A *Wasserstein barycenter* for a collection of distributions is a distribution that minimizes the sum of its squared Wasserstein distances to all the distributions in the collection. The barycenter is a counterpart of the arithmetic mean under the Euclidean distance for vectors, which also minimizes the total squared Euclidean distance. The Wasserstein barycenter produces geometrically meaningful "average" shapes, attracting growing attention in recent literature.

Its power has been demonstrated in applications to 2-D and 3-D shapes (Benamou et al., 2015; Solomon et al., 2015). The Wasserstein barycenter of the first-tier signatures provides the first-tier signature of a "mean" or "average" simulation. We can take this first-tier signature as a "skeleton" for the "mean" cloud simulation. In order to create an ensemble centroid that has cloud masses resembling those in the ensemble, we propose three schemes. They exploit the optimal transport between the barycenter and the first-tier signatures of individual cloud maps. Moreover, at the choice of the user, an ensemble can be treated as one group or multiple subgroups formed by D2-clustering (Li and Wang, 2008).

2.1 Mean Images by GEM

We now introduce the three schemes of deriving the mean image for a group of images. The detailed algorithms will be presented in Section 3.

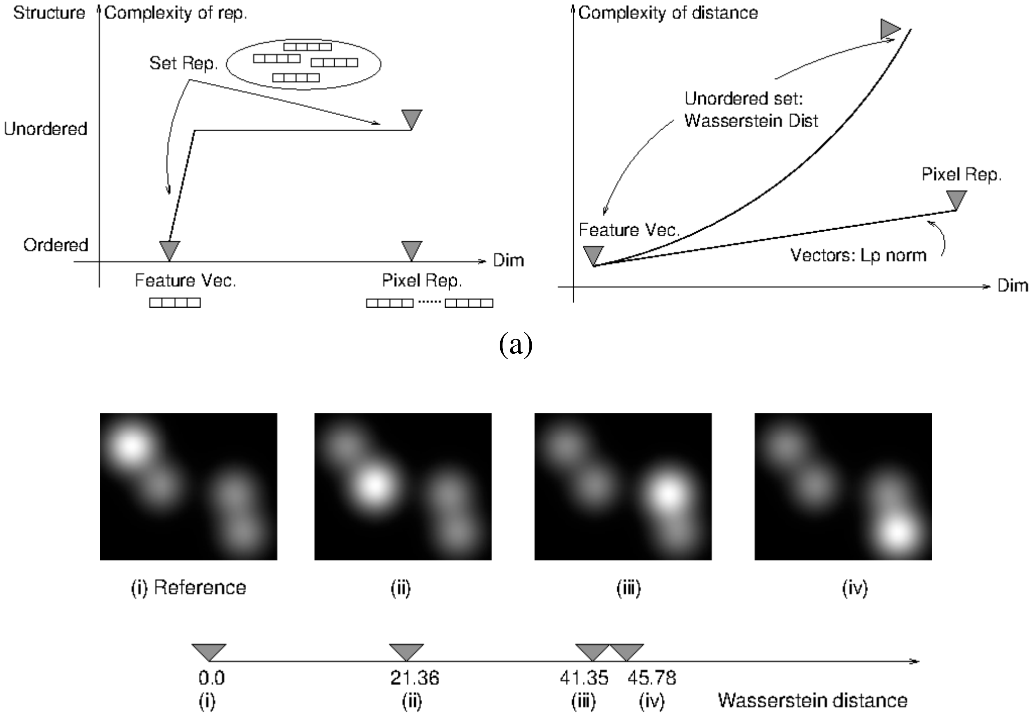
1. *Mixture Density Mean* (MDM) image: This is a cloud simulation that serves as a summary of all the cloud simulations in an ensemble. The first-tier signature of MDM is the Wasserstein barycenter of the first-tier signatures in the ensemble. The cloud mass at each location is then created by covariance fusion.

2. *Bayesian Posterior Mean* (BPM) image: We propose a Bayesian model that treats the aggregated mean image as an unobserved "true" image and the image given by any forecasting result as a random sample governed by a certain distribution. We propose schemes to set up the prior and conditional distributions. The "true" image is then estimated by the Bayesian posterior mean.

3. *In-sample Mean under Rigid Motion* (IM-RM) image: Given a set of instances and a centroid instance, which may or may not be a member of the set, the *in-sample mean* is the instance from the set that is closest to the centroid. For instance, if the centroid of a group of vectors is their arithmetic mean, then the in-sample mean is the vector closest to the arithmetic mean.

   For each cloud image in the ensemble, we find the optimal rotation and translation, called rigid motion together, that align its first-tier signature with the Wasserstein barycenter. We call the rigid motion solved by this method *Wasserstein Barycenter Guided Rigid Motion* (WB-RM). We transform each cloud map by WB-RM. The IM-RM image is the in-sample mean of all the transformed cloud maps when the MDM image is taken as the centroid. The purpose is to provide an average cloud map with the appearance of a simulated image by a forecasting model.

We refer to the MDM, BPM, and IM-RM images as the *aggregated simulation*. Each of them can be used as the ensemble centroid/mean.



(a)

(b)

Figure 1: Illustration for the Wasserstein metric. (a) Compare the complexity of different types of image representations including feature-based vectors, pixel-based vectors, and the set of unordered and dynamic vectors under the Wasserstein metric; (b) Four examples of discrete distributions and their Wasserstein distances to the reference distribution (first on the left).

2.2 Wasserstein Distance and Barycenter

The Wasserstein distance is a true metric defined for general probability measures. A thorough treatment of this metric and its applications in probability theory are referred to Rachev (1985). The Wasserstein metric is well defined for distributions with different support points, an important difference from some popular distances between distributions such as Kullback-Leibler (KL) divergence. It corresponds well with our heuristics. When we assess the similarity between two cloud images, we tend to build a correspondence between the cloud patches across the two images and combine the distances between cloud patches into an overall one for the whole images.

Consider two discrete distributions $\mathcal{P}^{(a)} = \{(w_i^{(a)}, x_i^{(a)}), i = 1, ..., m_a\}$ and $\mathcal{P}^{(b)} = \{(w_j^{(b)}, x_j^{(b)}), j = 1, ..., m_b\}$, where $w_i^{(a)}$ (or $w_j^{(b)}$) is the probability assigned to support point $x_i^{(a)}$ (or $x_j^{(b)}$), $x_i^{(a)}, x_j^{(b)} \in R^d$ (R^d is the d-dimensional Euclidean space). Let $\mathcal{J}_a = \{1, ..., m_a\}$, $\mathcal{J}_b = \{1, ..., m_b\}$. Denote the L_p norm by $\| \cdot \|_p$. Denote the Wasserstein metric under L_p norm between $\mathcal{P}^{(a)}$ and $\mathcal{P}^{(b)}$ by $W_p(\mathcal{P}^{(a)}, \mathcal{P}^{(b)})$. Then

$$\begin{aligned} \left(W_p(\mathcal{P}^{(a)}, \mathcal{P}^{(b)})\right)^p \quad &:= \min_{\{\pi_{i,j} \geqslant 0\}} \sum_{i \in \mathcal{J}_a, j \in \mathcal{J}_b} \pi_{i,j} \| x_i^{(a)} - x_j^{(b)} \|_p^p \ , \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{m_a} \pi_{i,j} = w_j^{(b)}, \ \forall j \in \mathcal{J}_b \\ & \textstyle\sum_{j=1}^{m_b} \pi_{i,j} = w_i^{(a)}, \ \forall i \in \mathcal{J}_a \ . \end{aligned} \tag{1}$$

It is proved that the above definition is a true metric under any L_p norm, where $p \geq 1$. We usually use the L_2 norm. For brevity of notation, we will simply denote W_2 as W in the rest of the paper. We call $\{\pi_{i,j}\}$ the *matching weights* or the *optimal transport*. Denote the optimal transport between $\mathcal{P}^{(a)}$ and $\mathcal{P}^{(a)}$ by $\Pi(\mathcal{P}^{(a)}, \mathcal{P}^{(b)}) = (\pi_{i,j})_{i \in \mathcal{J}_a, j \in \mathcal{J}_b}$.

In Figure 1 (a), we illustrate the complexity of a set representation and the complexity of Wasserstein distance defined on weighted sets. In terms of the amount of quantities contained in the representation, a set representation lies between a single feature vector and a pixel representation, and can reach either of the two extreme cases. We denote loosely in the figure the number of quantities in a representation as dimension although this terminology is not accurate for the set representation. In terms of the mathematical structure, both the feature vector and the pixel representation are ordered, meaning that any quantity is associated with a dimension of fixed meaning. However, a set representation comprises unordered weighted vectors. Both the weight and the vector itself vary across instances and have to be stored. The left panel in the figure compares the complexity of several distances. The computational cost of L_p norm increases linearly with dimension. However, the Wasserstein distance has no closed form solution and is solved by linear programming (LP). If standard LP numerical algorithms are applied, the complexity grows at least in polynomial orders on average. Designing scalable algorithms to solve the Wasserstein distance is an active research area. Even the state-of-the-art algorithms have computational complexity higher than that of L_p norm by several orders of magnitude.

Nevertheless, with the computational power of current typical desktop computers and the recent advance on numerical algorithms, statistical/machine learning methods based on the Wasserstein distance are attracting increasing attentions (Cuturi and Doucet, 2014; Rabin et al., 2011).

One appeal of the Wasserstein metric in contrast to popular distances such as L_2 (Euclidean distance) based on histograms or KL divergences is that it takes into account the geometric relationship between the support points. We illustrate this by an example in Figure 1 (b). Consider four discrete distributions with a common support set: $x_1 = (30.0, 28.0)^t$, $x_2 = (65.0, 54.0)^t$, $x_3 = (72.0, 120.0)^t$, and $x_4 = (107.0, 128.0)^t$. The distributions differ in the probabilities assigned to each support point. For every distribution, one support point has probability 0.43, while the other three have probability 0.19. For distribution $\mathcal{P}^{(i)}$ of random vector $X \in \mathcal{R}^2$, $P(X = x_i) = 0.43$, $i = 1, ..., 4$. Let us use $\mathcal{P}^{(1)}$ as the reference distribution. Obviously, under either the L_2 norm for the vector of probabilities on the support points or the KL divergence, the distance between $\mathcal{P}^{(i)}$, $i = 2, 3, 4$, and $\mathcal{P}^{(1)}$ is a constant. For instance, the KL divergence $D(\mathcal{P}^{(1)} \| \mathcal{P}^{(i)}) = 0.196$, $i = 2, 3, 4$. These two distances do not depend on the underlying distances between the support points. For better visualization, we show each distribution as an image. The pixel intensity is proportional to a Gaussian mixture density with four components each centered at one x_i. The prior for a component is the probability at x_i. The covariance matrix is spherical and identical across the components. Note that the Gaussian mixture density is used solely for visualization purpose and its covariance matrices for each component are irrelevant to the Wasserstein distances computed here. The Wasserstein distances are $W(\mathcal{P}^{(1)}, \mathcal{P}^{(2)}) = 21.36$, $W(\mathcal{P}^{(1)}, \mathcal{P}^{(3)}) = 41.35$, $W(\mathcal{P}^{(1)}, \mathcal{P}^{(2)}) = 45.78$. An inspection of the images in Figure 1 (b) suggests that these distance values reflect better our heuristics about the closeness of the images.

The Wasserstein barycenter for a collection of distributions is the counterpart of the arithmetic mean for vectors. Under the Euclidean distance, the arithmetic mean minimizes the total squared distance to all the vectors. Consider a discrete distribution $\mathcal{Q}$ with a pre-selected support size $\tilde{m}$: $\mathcal{Q} = \{(\pi_i, x_i), i = 1, ..., \tilde{m}\}$. The Wasserstein barycenter minimizes the total squared Wasserstein distances to the members of a set of distributions $\{\mathcal{P}^{(1)}, ..., \mathcal{P}^{(N)}\}$:

$$\min_{\mathcal{Q}} \sum_{i=1}^{N} W^2(\mathcal{Q}, \mathcal{P}^{(i)}) \ . \tag{2}$$

The above Wasserstein barycenter problem can be extended to a clustering problem using the commonly adopted principle of minimum within cluster variation. Suppose we cluster the distributions $\mathcal{P}^{(l)}$'s into K clusters each with a centroid distribution $\mathcal{Q}^{(k)}$, $k = 1, ..., K$. The optimization problem

$$\min_{\mathcal{Q}^{(1)},...,\mathcal{Q}^{(K)}} \sum_{i=1}^{N} \min_{k=1,...,K} W^2(\mathcal{Q}^{(k)}, \mathcal{P}^{(i)})$$

is called D2-clustering by Li and Wang (2008). It is mathematically challenging to solve the Wasserstein barycenter and even more so D2-clustering. We use the Accelerated D2 (AD2)-Clustering algorithm of Ye et al. (2017).

The phrase "barycenter" was coined by Agueh and Carlier (2011), who established the existence, uniqueness, and other theoretical properties of 2nd-order Wasserstein barycenter for continuous measures in the Euclidean space. For the special case of discrete distributions with finite but dynamic support sets, an algorithm to compute the barycenter was developed earlier by Li and Wang (2008), but that algorithm has strong limitation in scalability. Much effort has been devoted to the development of computationally efficient algorithms for Wasserstein barycenters in recent years, e.g., (Cuturi and Doucet, 2014; Benamou et al., 2015; Ye et al., 2017). A series of interesting work related to Wasserstein distance and barycenter has appeared in statistics literature recently. Minsker et al. (2014) and Srivastava, Li, and Dunson (2015) developed divide-and-conquer strategies for Bayesian inference on big data. Specifically, posterior distributions estimated from subsets of data are combined through their median distribution (e.g., in the Wasserstein metric space) (Minsker et al., 2014) or barycenter (Srivastava, Li, and Dunson, 2015). Li, Srivastava, and Dunson (2017) applied Wasserstein barycenter to estimate a posterior interval for massive data. Carlier, Chernozhukov, and Galichon (2016) extended quantile regression to vector-valued response using optimal transport. Sommerfeld and Munk (2016) derived the asymptotic distribution of empirical Wasserstein distances. Our purpose of using Wasserstein barycenter is different from the aforementioned works; and we use the state-of-the-art algorithm by Ye et al. (2017) with complexity linear in the data size.

3 Algorithms

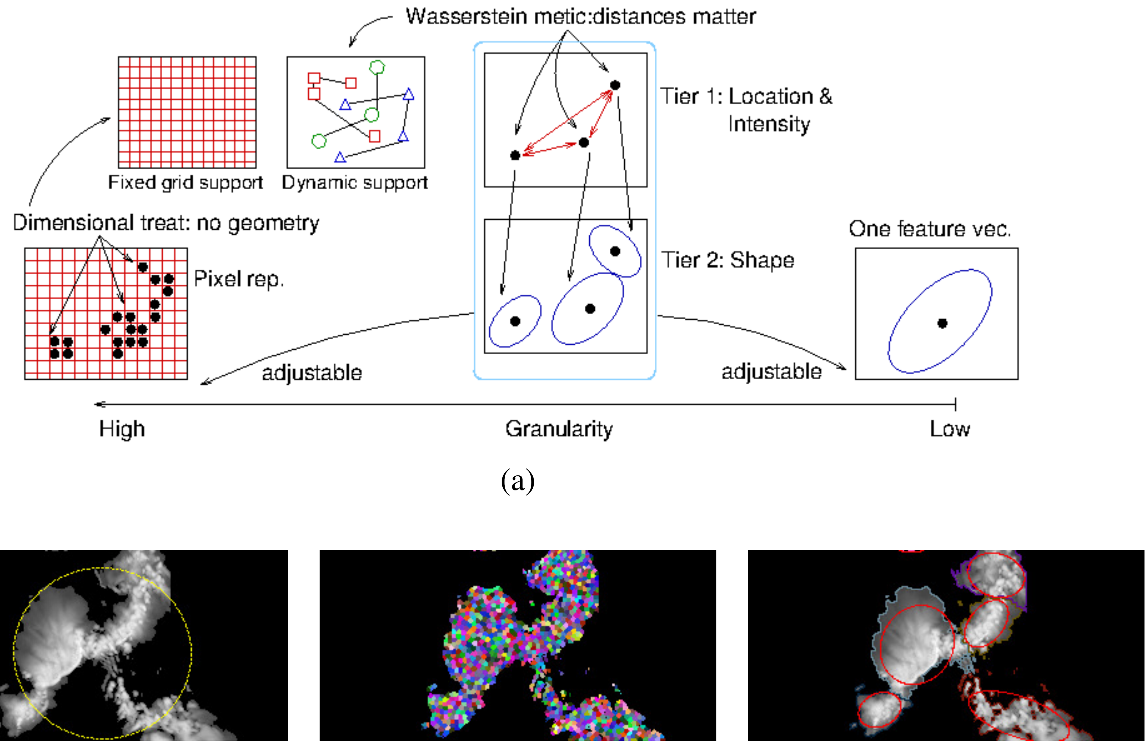
Suppose an image contains $n_{\text{v}} \times n_{\text{h}}$ pixels, where n_{v} is the number of rows and n_{h} is the number of columns in the image. A grayscale digital image is essentially a matrix of size $n_{\text{v}} \times n_{\text{h}}$, where each element records the value of the pixel at the corresponding position in the image plane. In our case, the value of a pixel is the cloud intensity at this pixel. Denote an entire image by $\mathcal{I}$. For a pixel with spatial coordinates $z = (z_{\text{v}}, z_{\text{h}})$ where $z_{\text{v}} = 0, 1, ..., n_{\text{v}} - 1$ is the vertical position and $z_{\text{h}} = 0, 1, ..., n_{\text{h}} - 1$ is the horizontal position, we denote its value by $\mathcal{I}(z)$. We may define an image by specifying $\mathcal{I}(z)$ at every z, as is done in Sections 3.2 and 3.4. In Section 3.3, we regard $\mathcal{I}$ as a random vector of dimension $n_{\text{v}} \times n_{\text{h}}$, which is formed by stacking the rows of the image matrix. It is then unnecessary to define $\mathcal{I}(z)$ for individual z's, and hence we will define $\mathcal{I}$ using matrix operations.

3.1 Two-tier Signature of Images

We emphasize that the motivation for the two-tier signature is to represent a cloud simulation at the object level so that cloud patches are extracted and specified directly by shape and location. Such high-level information is more meaningful than pixel intensities are for meteorologists whose primary attention is on patterns or characteristics of clouds. However, one pixel is very localized with respect to a cloud. The intensity at a pixel often varies widely when the weather forecasting model or parameters change. As a result, a straightforward pixel-wise average tends to eliminate important traits of clouds, which will be demonstrated clearly by our experiments. In addition, being a much more compact representation of cloud simulation images, the two-tier signature offers computational advantage for certain algorithms, but its purpose is deeper than data reduction.

In Figure 2 (a), we illustrate the two-tier signature and compare it with the straightforward pixel representation of the image as well as a feature vector representation. The number of cloud patches corresponds to the granularity level of the first-tier representation. At one extreme, when each cloud pixel is treated as one cloud patch, we obtain essentially the pixel representation, in which case the cloud shapes become meaningless for individual pixels. In contrast, in the two-tier signature, some features describe the shape directly. At the other extreme, the entire cloud image is characterized by one feature vector containing shape variables. Although one feature vector per instance is the standard data model in machine learning, it is inadequate to capture the complexity in the shapes of the clouds.

A pixel representation is usually treated as a high dimensional vector. Every pixel corresponds to one dimension, and the cloud intensity at the pixel is its value. The underlying meanings of the dimensions play no role in the analysis. For instance, in our case, the pixels are all from one image plane and have intrinsic geometric relationship, but this relationship is ignored in a dimensional treatment. In contrast, the first-tier signature is essentially a discrete distribution over the cloud locations after normalizing the cloud intensities on each patch. The cloud locations (i.e., the support points of the distribution) are dynamic and stored as part of the signature (see Figure 2 (a)).



(a)

(b)

Figure 2: Two-tier image signature. (a) An illustration for how the two-tier representation of cloud simulation images captures the geometry of the clouds without imposing strong assumptions, and why it is a unified framework under which the pixel-wise and feature vector representations are extreme cases. (b) An example to show the process of signature extraction. Left: Original image and the single Gaussian fitted; Middle: The cells generated which are marked by different colors; Right: The segmented clouds marked by boundaries of different colors and the fitted Gaussians to each of the segmented clouds.

We use the hierarchical mode association clustering (HMAC) algorithm by Li, Ray, and Lindsay (2007) to segment clouds. HMAC clusters data based on modes of a fitted kernel density. It merges data in a hierarchical fashion by gradually increasing the kernel bandwidth. It is found that HMAC is robust for non-Gaussian shape clouds and guarantees strong separation between clusters. Example applications of HMAC include Ray and Pyne (2012). The granularity of the segments can be easily adjusted by changing the kernel bandwidth. We create a signature for an image by the following process.

1. We first convert the image into a weighted set of pixel coordinates. For any pixel with positive cloud intensity, the horizontal and vertical coordinates of the pixel in the image plane are treated as a two-dimensional data point in the set and the cloud intensity at the pixel is taken as the weight assigned to this point. Clearly, we can fully reconstruct the image based on this data set. Denote this data set by $\mathcal{X} = \{x_1, x_2, ..., x_n\}$, where n is the number of non-zero pixels and $x_i = (x_{i,1}, x_{i,2})^t$, called *pixel location*, contains the vertical coordinate $x_{i,1}$ and horizontal coordinate $x_{i,2}$. Note that vectors are always assumed to be column vectors in this paper. Each pixel location x_i is associated with a weight w_i that equals the cloud intensity at this pixel.

2. For the sake of computational efficiency, we first summarize $\mathcal{X}$ by a large set of *cells*. Here, a *cell* is a group of pixels that are closely positioned, and it is acceptable not to distinguish pixels inside the cell. In another word, we quantize $\mathcal{X}$ so that clustering is later applied to a smaller set of representative points instead of the original points. The number of representative points is still greater than the possible number of cloud patches by one to several orders of magnitude. We employ the weighted k-means, also known as the weighted Lloyd algorithm in vector quantization, to divide the points into the cells. The *centroid* (also referred to as *codewords*) of each cell is defined as the weighted average of pixel positions. Suppose $\mathcal{X}$ are partitioned into n' cells. Denote the center of each cell by x'_i, $i = 1, ..., n'$. Again, x'_i is associated with weight w'_i, where w'_i is the sum of the weights of the points in cell i. Let $\mathcal{X}' = \{x'_1, x'_2, ..., x'_{n'}\}$ and $\mathcal{W}' = \{w'_1, ..., w'_{n'}\}$.

3. Apply 2-level HMAC to $\mathcal{X}'$ with weights $\mathcal{W}'$. The bandwidths used in HMAC determine the final number of clusters. We treat each cluster as one *cloud segment*, also called *cloud patch*. Once the cluster label of each cell is determined, the pixels in the cell will inherit its label. We thus obtain a segmentation of the pixels.

4. Suppose K cloud segments have been generated. Let $\mathcal{C}_k$ be the set of indices for x_i's that are contained in the kth cluster. For each cloud segment k, $k = 1, ..., K$, we extract the following summary information.

   (a) The total cloud intensity $s_k = \sum_{i \in \mathcal{C}_k} w_i$.

   (b) The weighted center of the pixels $\mu_k = \sum_{i \in \mathcal{C}_k} w_i x_i / s_k$.

   (c) The weighted covariance matrix of the pixels $\Sigma_k = \sum_{i \in \mathcal{C}_k} w_i (x_i - \mu_k)^t (x_i - \mu_k) / s_k$.

We denote the two-tier signature of the lth image by $\mathcal{M}^{(l)} = \{\mathcal{P}^{(l)}, \mathcal{G}^{(l)}\}$, where $\mathcal{P}^{(l)}$ is the first-tier signature and $\mathcal{G}^{(l)}$ is the second-tier signature. For the notations s_k, μ_k, and Σ_k, we add superscript (l) to indicate those quantities for the lth image. Suppose there are m_l cloud patches in the lth image. Let $\breve{s}^{(l)} = \sum_{j=1}^{m_l} s_j^{(l)}$ be the total sum of cloud intensity and $\tilde{s}_j^{(l)} = s_j^{(l)}/\breve{s}^{(l)}$. Then $\mathcal{P}^{(l)} = \{(\tilde{s}_j^{(l)}, \mu_j^{(l)}), j = 1, ..., m_l\}$ and $\mathcal{G}^{(l)} = \{\Sigma_j^{(l)}, j = 1, ..., m_l\}$. The two-tier signature of an image is essentially a Gaussian mixture model (GMM) (see Banfield and Raftery (1993); Melnykov and Maitra (2010) for survey on GMM). Our usage of GMM here is somewhat unusual. Instead of using GMM as a tool to model data distribution, we use it as a representation of a single object in an unsupervised learning framework. We exploit its capacity to retain the geometric characteristics of an image in a compact form.

We now use an example to illustrate the above process. In Figure 2 (b), we show the original cloud image, the cells generated, and the modal clustering result. We visualize the signature of the cloud image in the right panel of the figure. Each ellipse, corresponding to one cloud cluster, is centered at the center location of the cluster, and its orientation and size are determined by the covariance matrix of pixels in the cluster. Specifically, the long and short major axis of the ellipse are given by the first and second principal component directions computed from the covariance matrix. The radius of the ellipse is $1.65\sqrt{\lambda_i}$, $i = 1, 2$, where λ_i is the ith eigenvalue of the covariance matrix. Geometrically, $\sqrt{\lambda_i}$ is the standard deviation of the data projected onto the ith principal component direction. If we approximate the distribution of the projected data by a Gaussian, roughly 90% of the data will deviate from the mean by no more than $1.65\sqrt{\lambda_i}$. The yellow ellipse in the image at the left shows the contour of a single Gaussian distribution fitted on the cloud intensities. Clearly, given the non-elliptical shape of the cloud and the scattering of multiple patches of clouds in one image, the two-tier signature captures the cloud shape significantly better than a single fitted Gaussian.

3.2 Mixture Density Mean Image based on Wasserstein Barycenters

To generate the Mixture Density Mean (MDM) image as a summary of the cloud simulations in an ensemble, we first solve the Wasserstein barycenter for the first-tier signatures $\mathcal{P}^{(l)}$, $l = 1, ..., N$, in the ensemble. We then aggregate the shapes of the cloud patches using the second-tier signature $\mathcal{G}^{(l)}$. Denote the barycenter distribution by $\mathcal{Q}$. Let $\mathcal{Q} = \{(\alpha_j, \mu_j^*), j = 1, ..., \bar{m}\}$, where α_j is the probability on support point μ_j^*. The steps are detailed below.

1. Let the average number of cloud patches (rounded to the nearest integer) in an image be $\bar{m} = \left[\sum_{l=1}^{N} m_l/N\right]$. We set the support size of the barycenter $\mathcal{Q}$ to $\bar{m}$. We use the modified Bregman-ADMM (Alternating Direction Method of Multipliers) algorithm recently developed by Ye et al. (2017) to compute the barycenter $\mathcal{Q}$.

2. We also compute the optimal transport (matching weights) between $\mathcal{P}^{(l)}$ and $\mathcal{Q}$: $\Pi(\mathcal{P}^{(l)}, \mathcal{Q}) = (\pi_{i,j}^{(l)})$, $i = 1, ..., m_l$, $j = 1, ..., \bar{m}$. Recall that the covariance matrix for the ith cloud patch in the lth image is $\Sigma_i^{(l)}$ and the total cloud intensity in this patch is $s_i^{(l)}$. Denote the covariance matrix for the jth aggregated cloud patch by $\bar{\Sigma}_j$, $j = 1, ..., \bar{m}$. Define $\bar{\Sigma}_j = \dfrac{1}{N}\sum_{l=1}^{N} \dfrac{\sum_{i=1}^{m_l} \pi_{i,j}^{(l)}\Sigma_i^{(l)}}{\sum_{i=1}^{m_l} \pi_{i,j}^{(l)}}$. The rationale for the definition is to integrate all the cloud patches in all the images that have been matched to the jth cloud patch in the barycenter. As optimal transport generates a soft matching specified by the weights $\pi_{i,j}^{(l)}$ for the ith cloud patch in the lth image, we define $\bar{\Sigma}_j$ by the above weighted average.

3. Denote the MDM image by $\mathcal{I}_{\text{MDM}}$. Denote the density function of a Gaussian distribution by $\phi(\cdot)$. Let $f(x) = \sum_{i=1}^{\bar{m}} \alpha_i \phi_i(x \mid \mu_i^*, \bar{\Sigma}_i)$ be the mixture density of Gaussian distributions, where μ_i^* is the ith support point of the barycenter distribution $\mathcal{Q}$ and α_i is its probability. Suppose the image has n_{v} rows and n_{h} columns. Recall that the total cloud intensity for the lth image is $\breve{s}^{(l)} = \sum_{i=1}^{m_l} s_i^{(l)}$. Denote their mean by $\bar{s} = \frac{1}{N}\sum_{l=1}^{N} \breve{s}^{(l)}$. Consider a pixel with spatial coordinates $(z_{\text{v}}, z_{\text{h}})$ where z_{v} is the vertical position, $z_{\text{v}} = 0, 1, ..., n_{\text{v}} - 1$, and z_{h} is the horizontal position, $z_{\text{h}} = 0, ..., n_{\text{h}} - 1$ in $\mathcal{I}_{\text{MDM}}$. Then the pixel intensity of $\mathcal{I}_{\text{MDM}}$ at $z = (z_{\text{v}}, z_{\text{h}})$ is given by

$$\mathcal{I}_{\text{MDM}}(z) = \frac{\bar{s} f(z)}{\sum_{z' \in [0,...,n_{\text{v}}] \times [0,...,n_{\text{h}}]} f(z')} .$$

   The normalization above ensures that the total cloud intensity in $\mathcal{I}_{\text{MDM}}$ is the same as the average total cloud intensity of all the images.

We have so far assumed that we want to aggregate all the cloud images in an ensemble into one centroid image. If we believe that the cloud images fall into several subgroups and would like to aggregate each group separately, we can first cluster the images using the D2-clustering algorithm. We again use the fast algorithm of Ye et al. (2017).

3.3 Bayesian Posterior Mean Image

Denote an ensemble of images by $\{\mathcal{I}_1, ..., \mathcal{I}_N\}$. Let the number of pixels in the images be $n_{\text{v}} \times n_{\text{h}}$. Consider each image as a vector formed by stacking the pixel intensities in the whole image. We propose the following Bayesian model for estimating the true image $\mathcal{I}^*$. We assume that $\mathcal{I}_1$, ..., $\mathcal{I}_N$ are conditionally independent given $\mathcal{I}^*$, and the conditional density $\mathcal{I}_i \mid \mathcal{I}^*$ is multivariate Gaussian with mean vector equal to $\mathcal{I}^*$ and covariance matrix equal to $\sigma_i^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. In addition, we let the prior on $\mathcal{I}^*$ be multivariate normal $\mathcal{N}_{n_{\text{v}}\times n_{\text{h}}}(\mathcal{I}^{(0)}, \sigma_0^2\mathbf{I})$.

In order to choose the parameters in the model, we follow an empirical Bayesian spirit. In particular, we take $\mathcal{I}_{\text{MDM}}$ which is computed from the data as $\mathcal{I}^{(0)}$. Recall that the first-tier signature of image $\mathcal{I}_i$ is denoted by $\mathcal{P}^{(i)}$, and the barycenter of $\mathcal{P}^{(i)}$'s is $\mathcal{Q}$. We let $\sigma_i^2 \propto W^2(\mathcal{Q}, \mathcal{P}^{(i)})$, $i = 1$, ..., N, and $\sigma_0^2 \propto \frac{1}{N}\sum_{i=1}^N W^2(\mathcal{Q}, \mathcal{P}^{(i)})$. It can be shown that the posterior distribution of $\mathcal{I}^*$ given $\{\mathcal{I}_1, ..., \mathcal{I}_N\}$ is Gaussian with posterior mean $E(\mathcal{I}^* \mid \mathcal{I}_1, ..., \mathcal{I}_N) = w_0\mathcal{I}_{\text{MDM}} + \sum_{i=1}^{N} w_i\mathcal{I}_i$, where the weights $w_i \propto \frac{1}{\sigma_i^2}$, $i = 0, ..., N$, and $\sum_{i=0}^N w_i = 1$. We use the above posterior mean as the Bayesian posterior mean (BPM) image, $\mathcal{I}_{\text{BPM}}$, for an ensemble. The w_i's can be calculated by knowing the σ_i^2's up to a constant. Specifically,

$$\mathcal{I}_{\text{BPM}} = w_0\mathcal{I}_{\text{MDM}} + \sum_{i=1}^{N} w_i\mathcal{I}_i \,,$$

where $w_0 = \dfrac{\frac{N}{\sum_{j=1}^N W^2(\mathcal{Q},\mathcal{P}^{(j)})}}{\frac{N}{\sum_{j=1}^N W^2(\mathcal{Q},\mathcal{P}^{(j)})} + \sum_{j=1}^N \frac{1}{W^2(\mathcal{Q},\mathcal{P}^{(j)})}}$, $w_i = \dfrac{\frac{1}{W^2(\mathcal{Q},\mathcal{P}^{(i)})}}{\frac{N}{\sum_{j=1}^N W^2(\mathcal{Q},\mathcal{P}^{(j)})} + \sum_{j=1}^N \frac{1}{W^2(\mathcal{Q},\mathcal{P}^{(j)})}}$.

The posterior covariance matrix of $\mathcal{I}^*$ given $\{\mathcal{I}_1, ..., \mathcal{I}_N\}$ is $\sigma_{\text{BP}}^2\mathbf{I}$, where σ_{BP}^2 is determined by $\dfrac{1}{\sigma_{\text{BP}}^2} = \displaystyle\sum_{i=0}^{N} \frac{1}{\sigma_i^2}$. The estimation for σ_{BP}^2, denoted by $\hat{\sigma}_{\text{BP}}^2$, can be obtained by estimating σ_i^2 first. Once we have estimated $\mathcal{I}^*$ by $\mathcal{I}_{\text{BPM}}$, we can estimate σ_i^2, $i = 1, ..., N$, by $\hat{\sigma}_0^2 = \dfrac{\|\mathcal{I}_{\text{BPM}} - \mathcal{I}_{\text{MDM}}\|^2}{n_{\text{v}} \times n_{\text{h}}}$ and $\hat{\sigma}_i^2 = \dfrac{\|\mathcal{I}_{\text{BPM}} - \mathcal{I}_i\|^2}{n_{\text{v}} \times n_{\text{h}}}$.

3.4 In-Sample Mean under Rigid Motion

An MDM image is synthesized by modeling each cloud patch by a Gaussian distribution over the pixel positions. The clouds look much smoother than those in the ensemble members. If we want to have an aggregated image resembling an ensemble member at high granularity, a natural idea is to use an existing member closest to a mean/centroid image in some sense. We call such a member an in-sample mean. Here we take the MDM image as the centroid. Because rigid motion of translation and rotation does not change the appearance of clouds, we allow rigid motion to be applied before comparing an ensemble member with the centroid image. We call this image the In-sample Mean under Rigid Motion (IM-RM). Figure 3 provides the flow chart for generating the IM-RM image.

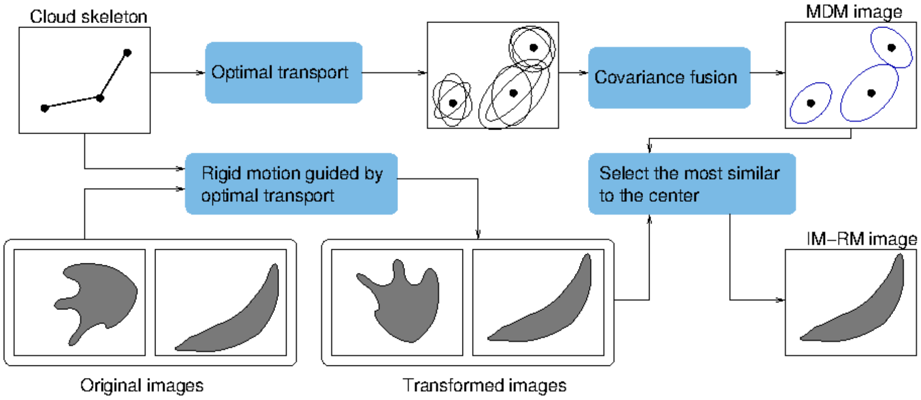


Figure 3: The flow chart for generating the IM-RM aggregated cloud simulation image.

We first optimize the rigid motion to be applied to any ensemble member by the Wasserstein Barycenter Guided Rigid Motion (WB-RM) algorithm described below.

1. Consider the first-tier signature $\mathcal{P}^{(l)} = \{(\tilde{s}_i^{(l)}, \mu_i^{(l)}), i = 1, ..., m_l\}$ of the lth image, $l = 1, ..., N$, and its optimal transport to the barycenter $\mathcal{Q} = \{(\alpha_j, \mu_j^*), j = 1, ..., \bar{m}\}$: $\Pi(\mathcal{P}^{(l)}, \mathcal{Q}) = (\pi_{i,j}^{(l)})$, $i = 1, ..., m_l$, $j = 1, ..., \bar{m}$. We compute the transported $\mu_i^{(l)}$ according to $\Pi(\mathcal{P}^{(l)}, \mathcal{Q})$ by $\tilde{\mu}_i = \sum_{j=1}^{\bar{m}} \pi_{i,j}^{(l)} \mu_j^* / \sum_{j=1}^{\bar{m}} \pi_{i,j}^{(l)}$, $i = 1, ..., m_l$.

2. We then optimize over a rotation matrix $\mathbf{R}$ and a translation ζ such that the transformed $\mu_i^{(l)}$, $i = 1, ..., m_l$ are closest to $\tilde{\mu}_i$, $i = 1, ..., m_l$ in an overall sense, that is, to solve $\underset{\mathbf{R},\zeta}{\operatorname{argmin}} \sum_{i=1}^{m_l} \tilde{s}_i^{(l)} \|(\mathbf{R}\mu_i^{(l)} + \zeta) - \tilde{\mu}_i\|^2$, where $\| \cdot \|$ is the L_2 norm, and all the vectors are column vectors. The algorithm of Sorkine-Hornung and Rabinovich (2017) is used to solve $\mathbf{R}$ and ζ.

3. Apply the rotation $\mathbf{R}$ and translation ζ to every pixel in the image to obtain a new image. Let the coordinate vector in the image plane be $z = (z_{\text{v}}, z_{\text{h}})^t$ and the original image be $\mathcal{I}(z)$ and the image after the rigid motion be $\mathcal{I}_{\text{RM}}(z)$. Then $\mathcal{I}_{\text{RM}}(z) = \mathcal{I}(\mathbf{R}^t(z - \zeta))$.

Let the original images in the ensemble be $\mathcal{I}_l$, $l = 1, ..., N$. We use the above algorithm to obtain $\mathcal{I}_{\text{RM},l}$. We then compute the total pixel-wise squared distance between $\mathcal{I}_{\text{RM},l}$ and $\mathcal{I}_{\text{MDM}}$ and choose image l^* that is closest to $\mathcal{I}_{\text{MDM}}$ according to this distance. We define the IM-RM image $\mathcal{I}_{IM-RM} = \mathcal{I}_{\text{RM},l^*}$.

Since the points under rotation and translation are two-dimensional, the rotation matrix $\mathbf{R}$ is determined by a rotation angle θ. The rigid motion parameters θ and ζ reflect how much motion is needed to align $\mathcal{P}^{(l)}$ with $\mathcal{Q}$. Their histograms can show the amount of variations within an ensemble.

3.5 Evaluation

To evaluate the mean images generated by different methods, we propose a distance that directly takes into account the shape, location, and intensity of cloud patches based on the two-tier signature. We will not use a pixel-wise average distance between two images because of the intrinsic limitations of the pixel-wise representation of cloud simulations, which are right at the heart of our motivation for developing the GEM system. Detailed discussions have been provided in Section 2 and 3.1.

As each cloud patch in the two-tier signature is characterized by a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ whose mean vector μ corresponds to the center location of the cloud and covariance matrix Σ reflects the shape, we use the Wasserstein distance between two Gaussian distributions as a measure of similarity between two cloud patches. For Gaussian distributions with density functions $\phi_1(x \mid \mu_1, \Sigma_1)$ and $\phi_2(x \mid \mu_2, \Sigma_2)$, the Wasserstein distance is given by a closed form (Givens and Shortt, 1984):

$$W^2(\phi_1, \phi_2) = \|\mu_1 - \mu_2\|^2 + tr\left(\Sigma_1 + \Sigma_2 - 2\left(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2}\right)^{1/2}\right) . \tag{3}$$

For two image signatures specified by GMMs $\mathcal{M}^{(i)}(x) = \sum_{j=1}^{m_i} \tilde{s}_j^{(i)} \phi_j^{(i)}(x \mid \mu_j^{(i)}, \Sigma_j^{(i)})$, $i = 1, 2$, we consider each as a discrete distribution with probabilities $(\tilde{s}_j^{(i)})_{j=1}^{m_i}$ over the parameter space of the mean vector and covariance matrix. Using the Wasserstein distance in Eq. (3) as the baseline distance in the parameter space, we can define a distance, denoted by $\tilde{W}(\mathcal{M}^{(1)}, \mathcal{M}^{(2)})$ based on optimal transport, in the same way as in Eq. (1):

$$\begin{aligned} \left(\tilde{W}(\mathcal{M}^{(1)}, \mathcal{M}^{(2)})\right)^2 \quad &:= \min_{\{\pi_{i,j} \geqslant 0\}} \sum_{i=1,\dots,m_1, j=1,\dots,m_2} \pi_{i,j} W^2(\phi_i^{(1)}, \phi_j^{(2)}) \ , \\ &\text{s.t.} \quad \textstyle\sum_{i=1}^{m_1} \pi_{i,j} = \tilde{s}_j^{(2)}, \ \forall j = 1, ..., m_2 \, ; \ \sum_{j=1}^{m_2} \pi_{i,j} = \tilde{s}_i^{(1)}, \ \forall i = 1, ..., m_1 \ . \end{aligned} \tag{4}$$

We call $\tilde{W}$ the *Minimized Aggregated Wasserstein (MAW)* distance. It was proposed by Chen, Ye, and Li (2016) as a computationally efficient approximation to the Wasserstein distance between two GMMs. MAW is particularly appealing for our evaluation purpose here because the distance between two Gaussian densities decomposes clearly into one part corresponding to location and the other to shape.

Our second approach to evaluate the mean images focuses on the fidelity of the distribution of the cloud intensities in an image, that is, marginal density of pixel values. The geometric information about clouds is filtered out by the marginal density. Thus this way of evaluation is complementary to MAW. Despite its restrictiveness, the marginal density captures important characteristics that meteorologists care about. For instance, the peak value of cloud intensities in an image is a major indicator of the strength of a weather system. In our experiments, we estimate the marginal density by normalized histograms. L_1 distance is then used to measure the difference between two histograms, each specified by a vector of frequencies over the histogram bins.

4 Experiments

We apply the GEM toolkit to two ensemble sets of simulated cloud images (referred to as Ensemble 1 and 2). Each set contains 41 ensemble members generated by the same numerical weather prediction model but with different initial conditions. More specifically, as detailed in Melhauser and Zhang (2012), the two-way nested fully compressible, non-hydrostatic Weather Research and Forecast (WRF) model (version 2.2; Skamarock et al. (2005)) is used to hindcast (apply forecasting to past events for reasons such as testing) the weather conditions valid at 0600 UTC 10 June 2003. The same model is initialized with 41 different initial conditions valid at 1200 UTC that include the mean estimated posterior uncertainties from the data assimilation experiment reported in Meng and Zhang (2008). The model used for this study employed 4 two-way nested domains with grid spacings of 90 km, 30km, 10km and 3.3km, respectively (see Figure 3 of Melhauser and Zhang (2012)). The two ensembles of cloud simulation images analyzed in this study were produced from the model that simulated radar reflectivity from the two innermost model domains. Ensemble 2 is in a larger domain with larger physical distances between pixels (10-km grid spacing) and Ensemble 1 is in a smaller subset domain but with smaller distances between pixels (3.3-km grid spacing). The higher the pixel value, the stronger or more severe the weather represented by the cloud is.

We first extract the two-tier signature for each image using the algorithm in Section 3.1. By using different kernel bandwidths in the HMAC segmentation algorithm, we obtained signatures at several levels of granularity. As explained in Li, Ray, and Lindsay (2007), when a larger Gaussian kernel bandwidth is used in density estimation, more Gaussian components tend to merge into the same mode, and thus fewer clusters are obtained (corresponding to lower granularity). The Gaussian kernel bandwidth used by HMAC is the only tuning parameter required to generate the synthetic mean image by any of the three schemes. We lack an objective criterion to decide this parameter based on the images themselves, which are the only information assumed by our current visualization system. On the other hand, because this parameter corresponds in a simple manner to the granularity of the cloud patches, we expect that users, primarily meteorologists, may be able to calibrate the parameter based on their knowledge about the appropriate sizes of clouds. In our experiments, for Ensemble 1, the average number of cloud patches in an image is $\bar{m} = 5, 9, 15, 50$; and for Ensemble 2, $\bar{m} = 5, 8, 15, 45$. The highest granularity levels correspond to $\bar{m} = 50$ and $\bar{m} = 45$, respectively. As we will see, the results obtained across a range of values of $\bar{m}$ are highly consistent.

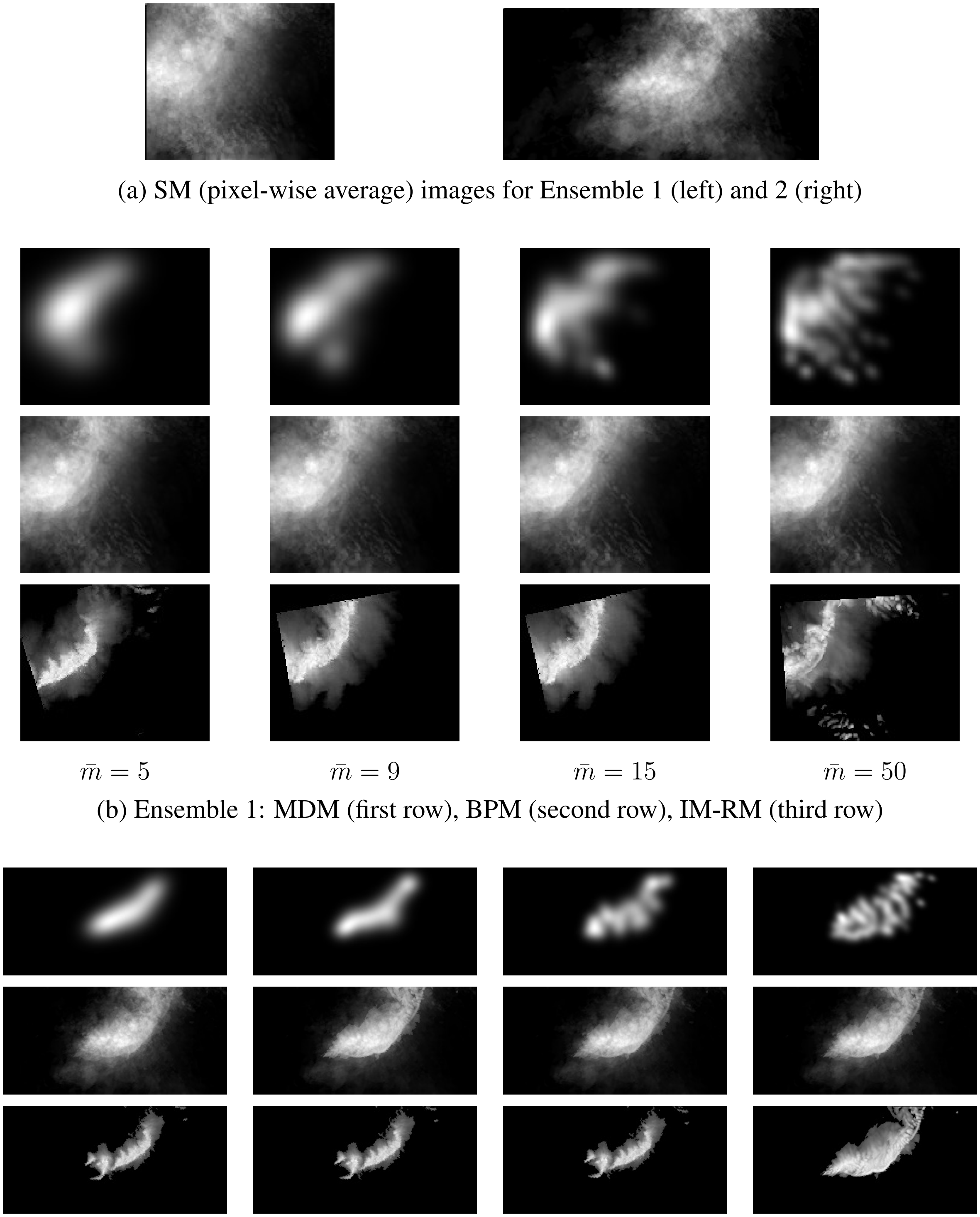
4.1 Visual and Quantitative Comparisons

At each granularity level, we obtained the MDM, BPM, and IM-RM aggregated images (also called mean images) for the whole ensemble. For all the cloud simulation images we show, the pixel values have been scaled to the maximum span from 0 to 255 (white). The real maximum cloud intensity values for the images (including the original ensemble members) are all below 60. We scaled the values for the clarity of illustration as the unscaled versions will appear quite dark.

For comparison, we show the pixel-wise average of the cloud simulations for each ensemble in Figure 4 (a). For brevity, we refer to the pixel-wise average as Simple Mean (SM) image which is mostly used for integrating cloud simulations in existing practice. The mean images by GEM are shown in Figure 4 (b) for Ensemble 1 and (c) for Ensemble 2. The MDM, BPM, and IM-RM images are shown in the first, second, and third rows respectively. When the barycenter support size increases, the MDM image shows more details, but the MDM images at different granularity levels are consistent in the sense that the positions and rough shapes of the clouds in the images are quite stable. Because the MDM images are simulated using a mixture of Gaussian distributions, they have a "smoothed out" appearance, the texture not resembling a typical simulation. This is not the case with the IM-RM images since they are the original images transformed by rigid motion. On the other hand, they are similar to the MDM images in the overall shapes. Both the MDM and IM-RM images indicate clearly the shapes of clouds, while the SM images only show rather smeared masses of cloud pixels. The BPM images are weighted average of the original images and are more similar to the MDM images than the SM images are.

To evaluate the methods numerically, we computed the MAW distance between the mean image of each scheme and every image in an ensemble. The average MAW distances within each ensemble are shown in Figure 5 (a). Each curve in a plot corresponds to one method applied at different granularity levels of the two-tier signature. As the MDM image is generated directly from a GMM, we use this GMM as its image signature. For the aggregated images obtained by SM, BPM, or IM-RM, we applied the same image signature extraction process as that applied to the individual ensemble images. In the cloud segmentation step, at any granularity level, we set the Gaussian kernel bandwidth to the same value used for the ensemble images. As shown in Figure 5 (a), for both ensembles, MDM achieves the lowest average distance across all the granularity levels, while SM yields the largest average distance. When the granularity level increases, the two-tier signature becomes closer to the pixel-wise representation of the image. As a result, we observe that the range of the distances obtained by different methods reduces.

In a quantitative term, the “smearing” effect of simple pixel averaging is captured by the remarkable drop in the peak cloud intensity in an image. As the positions of the peak value vary broadly in different images, after simple pixel-wise averaging, the maximum intensity can decrease dramatically. This problem is quite effectively avoided by our MDM and IM-RM methods. In Figure 5 (b), we show the boxplots of the maximum cloud intensity for several groups of images. First, to create a common ground for comparison, we obtained the boxplot for the maximum intensities of the original images in each ensemble (labeled as “Original” in the figure). For the SM image, there is a single value for each ensemble. For MDM, BPM, IM-FM respectively, the boxplot is for results obtained at the four granularity levels. As IM-RM images are rotated and shifted versions of the most representative images of each ensemble, the boxplot is closest to that of the original images. With MDM, the maximum intensities are reduced by a small amount. The maximum intensity of the SM image is much lower than the average of the original image, which is above 55.0 for both ensembles. For both ensembles, the maximum intensity of SM is below 20.0, while the smallest such value from the original images is

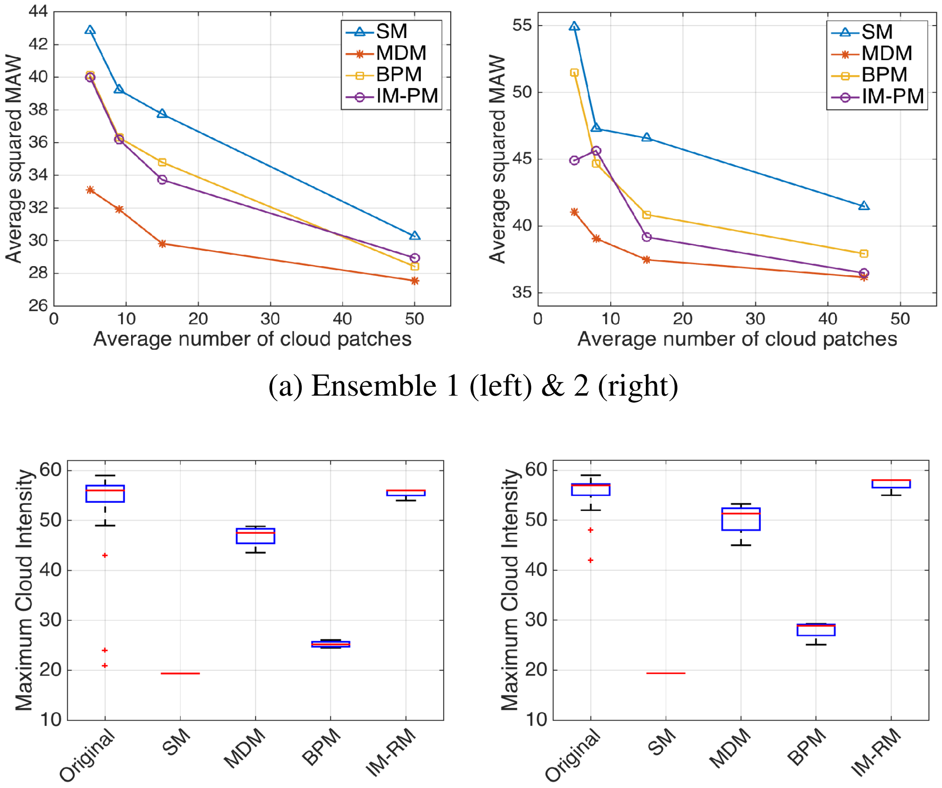


(a) SM (pixel-wise average) images for Ensemble 1 (left) and 2 (right)

(b) Ensemble 1: MDM (first row), BPM (second row), IM-RM (third row)

(c) Ensemble 2: MDM (first row), BPM (second row), IM-RM (third row)

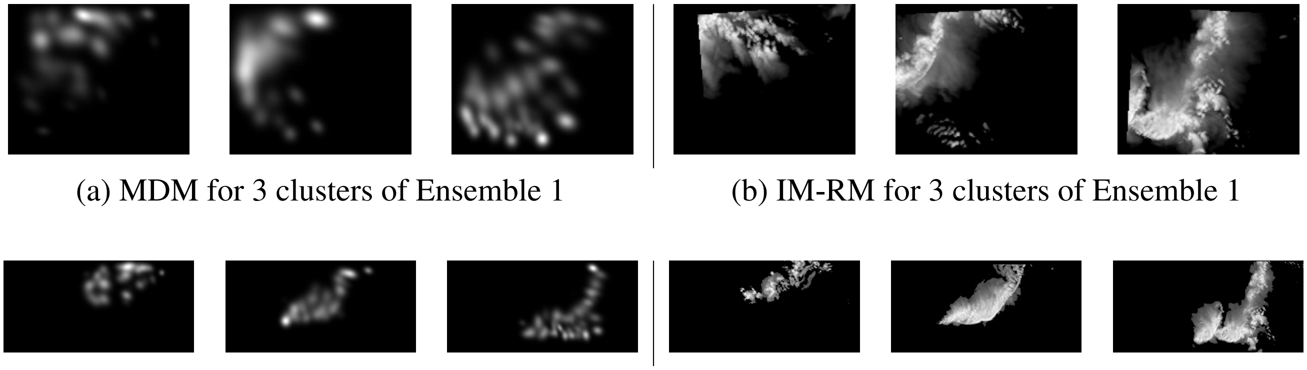
Figure 4: Comparison of SM images and mean images by GEM for Ensemble 1 and 2 at four granularity levels. The barycenter support size includes 5, 9, 15 and 50 for Ensemble 1 and 5, 8, 15 and 45 for Ensemble 2.



(a) Ensemble 1 (left) & 2 (right)

(b) Ensemble 1 (left) & 2 (right)

Figure 5: Quantitative comparison of mean images. (a): Average squared MAW distance between a mean image and all the images in the ensemble. Results are obtained across four granularity levels. For Ensemble 1, the average number of cloud patches across the granularity levels is 5, 9, 15, 50; for ensemble 2, 5, 8, 15, 45. (b): Boxplots for the maximum cloud intensity in each image. The boxplots are created for the following groups of images: all the members of the ensemble ("Original"), SM image for the ensemble, MDM, BPM, IM-RM images obtained at the four granularity levels respectively.



(a) MDM for 3 clusters of Ensemble 1

(b) IM-RM for 3 clusters of Ensemble 1

(c) MDM for 3 clusters of Ensemble 2

(d) IM-RM for 3 clusters of Ensemble 2

Figure 6: D2-Clustering results for Ensemble 1 and 2 at the highest granularity level. The average number of cloud patches for Ensemble 1 is 50 and that for Ensemble 2 is 45.

around 21.0 for Ensemble 1 and 42.0 for Ensemble 2. Albeit better than SM, the boxplot of BPM also differs significantly from that of the original images.

We compared the histograms of the cloud intensities in the mean images obtained by different methods. We also computed the histogram for every original image and use the average histogram across an ensemble as a benchmark (called "reference"). For both ensembles, the histograms show that MDM and IM-RM yield results similar to the reference, while those of BPM and SM are quite different. In Table 1, we report the L_1 distance between the reference histogram and those of the mean images. We also use the average distance between the reference histogram and that of each original image as a yardstick since it reflects the intrinsic variation within the ensemble. Denote the reference distance by D_{ref}. The distances achieved by MDM and IM-RM are either slightly higher than D_{ref} or considerably lower, the latter occurring much more often. For Ensemble 1, with only one exception, the distances by MDM or IM-RM are lower than half of D_{ref}. The results by BPM are clearly worse than MDM and IM-RM, but better than SM by a large margin.

	Ensemble 1				Ensemble 2			
D_{ref} (Ave. of orig.)	0.258				0.107			
SM	0.809				0.606			
	Ave. # cloud patches				Ave. # cloud patches			
Our methods	5	9	15	50	5	8	15	45
MDM	0.107	0.116	0.129	0.214	0.128	0.109	0.137	0.161
BPM	0.564	0.575	0.550	0.527	0.407	0.323	0.340	0.326
IM-RM	0.097	0.072	0.071	0.042	0.072	0.072	0.071	0.047

Table 1: Comparison of the L_1 distance between the normalized histograms. For each ensemble, the average histogram across the images in the ensemble is used as the reference histogram. We use D_{ref} to denote the average distance between the histogram of each ensemble member image and the reference. It reflects the variation within an ensemble. For each of our methods, results are reported for the mean images obtained at different levels of granularity (as reflected by the average number of cloud patches).

In summary, the most straightforward but least informative way of generating a mean image is by SM (equal weights on the ensemble members). One of the drawbacks of SM is that it diminishes the potential intensity (and thus the forecasted severe weather risk potential) of cloud patches, which vary widely in position and intensity among ensemble members. BPM has similar limitations although it improves SM by using unequal weights on different ensemble members. MDM, however, defines an average from a new angle. Instead of operating at the pixel-level, it computes the averages over attributes of segmented clouds. As demonstrated by the experiments, this strategy enables MDM to achieve high fidelity in terms of the cloud intensity distribution as well as the MAW distance. On the other hand, because the MDM image is not generated by a weather forecasting model but by object-level synthesis, it loses the sharpness of clouds. Being a true ensemble member (up to a rigid motion), the IM-RM image preserves the cloud sharpness. Although BPM is not the best by any of the quantitative measures, it is kin to the SM method, which is currently most used. We thus find it interesting to study. If a user is interested in the most representative scenario of the ensemble presented as a real individual forecast, IM-RM is most suitable. But if he or she is interested in the average object-level characteristics of the ensemble, MDM provides a more direct visualization.

4.2 Assessment of Uncertainty and Computational Efficiency

For the BPM images, we estimated the posterior variance of each pixel value $\hat{\sigma}^2_{\mathrm{BP}}$ (see Section 3.3). We then computed the ratio $\sqrt{\frac{\hat{\sigma}^2_{\mathrm{BP}}}{\|\mathcal{I}_{\mathrm{BPM}}\|^2/(n_{\mathrm{v}} \times n_{\mathrm{h}})}}$. The results across four granularity levels are provided in Table 2. When we computed the IM-RM image for an ensemble, we obtained the rotation angle and translation for each image. Take Ensemble 1 as an example. The rotation angle varies in a narrow range of around -15° to 15° with the highest percentages of images having near zero rotation. The average rotation angle is 0.55° with standard deviation 6.36°. The translation distance ranges from 0 to 80 pixels (the image size is 131×158). The average translation is 25.36 with standard deviation 14.65.

	Ensemble 1				Ensemble 2			
$\bar{m}$ (Ave. # cloud patches)	5	9	15	50	5	8	15	45
$\sqrt{\frac{\hat{\sigma}^2_{\mathrm{BP}}}{\|\mathcal{I}_{\mathrm{BPM}}\|^2/(n_{\mathrm{v}} \times n_{\mathrm{h}})}}$ (%)	15.4	15.3	15.0	14.5	19.7	17.6	18.0	17.6

Table 2: The estimated posterior variance for the BPM images across four granularity levels.

The vast majority of the computational time for generating the aggregated simulations is on computing the Wasserstein barycenters. We report the running time in Table 3. The running time is based on the Matlab implementation of the algorithm of Ye et al. (2017) on a single core Mac CPU of 3.5GHz. The computation time increases rapidly when the average support size of the signatures grows. The support size of the barycenter is set to be the average support size of the image signatures. When $\bar{m} = 5$, it takes about 0.7 seconds, while at $\bar{m} = 50$ for Ensemble 1, it takes about 348 seconds. The package we used can run on multiple cores to dramatically reduce the computation time. The reason for not experimenting with higher values of $\bar{m}$ is not that the algorithm of Ye et al. (2017) is too slow but that the cloud patches become overly localized when the granularity level is too high. In fact, the algorithm of Ye et al. (2017) is the state-of-the-art in terms of speed (linear in data size). Detailed discussion on the complexity of the algorithm is referred to that paper.

	Ensemble 1				Ensemble 2			
$\bar{m}$	5	9	15	50	5	8	15	45
Time (sec.)	0.77	1.44	3.56	348.13	0.72	1.29	5.53	177.29

Table 3: The running time of generating the Wasserstein barycenters versus the average support size of the distributions.

4.3 Clustering based on Wasserstein Distance

We have so far aggregated each ensemble to a single mean image. It is interesting to explore subgroups within the ensemble. We again applied the fast algorithm of Ye et al. (2017) to cluster the first-tier signatures into 3 groups. For each subgroup, we computed the MDM, BPM, and IM-RM images. The results obtained at the highest granularity level are shown in Figure 6. Due to lack of space, only MDM and IM-RM images are displayed. We can observe rather distinct patterns of clouds in the aggregated images for the three subgroups in each ensemble. For Ensemble 1, the three clusters contain 9, 20 and 12 images, respectively, and for Ensemble 2, they contain 13, 20 and 8 images, respectively. If we compare the results in Figure 4 and Figure 6, we see that the aggregated mean for the entire ensemble (last column of Figure 4 (b) and (c)) is a reasonable combination of the three subgroups (Figure 6). The grouping generated by D2-clustering helps us organize the many simulations in one ensemble so that we can more easily see the dominant differences in the cloud patterns.

5 Conclusions and Future Work

We have developed the GEM toolkit for summarizing an ensemble of simulated cloud images in order to solve the imperative issue of information overload faced by meteorologists. Viewed from a typical statistical perspective, our method can be considered as a new way of defining the mean of physical signals, e.g., images, such that the geometric nature of the signals can be preserved. To obtain geometrically meaningful average simulations, we propose to represent cloud maps by a two-tier data model and exploit the Wasserstein distance in clustering and centroid computation.

As aforementioned, the GEM system requires a given granularity level to generate the mean images. Due to the more or less subjective nature of visualization problems in general, it is difficult to choose the best granularity level according to any numerical measure. Our results, as shown by the example images and numerical evaluation, are stable over the different levels. We anticipate that in its deployment, the system will be calibrated by users with extra domain information.

One interesting future direction is to take into account the covariance matrix of a Gaussian component during the optimization of the barycenter. Specifically, recall that the two-tier signature of the lth image is essentially a GMM $\mathcal{M}^{(l)} = \{\mathcal{P}^{(l)}, \mathcal{G}^{(l)}\}$, where $\mathcal{P}^{(l)}$ is the discrete distribution over the locations and $\mathcal{G}^{(l)}$ contains the covariance matrices that capture the shapes of cloud patches. Instead of solving the barycenter according to Eq. (2), an alternative approach would be to solve a generalized barycenter by $\min_{\mathcal{Q}} \sum_{l=1}^{N} \tilde{W}^2(\mathcal{Q}, \mathcal{M}^{(l)})$, where $\tilde{W}$ is the MAW distance defined in Eq. (4) and $\mathcal{Q}$ is a GMM with a given number of components. This new optimization formulation is appealing as it treats cloud location and shape in a more unified way. On the other hand, to solve the new barycenter problem, we must extend the algorithm of Ye et al. (2017). Whether the new formulation achieves better results and how much the computational cost is need to be examined through experiments.

6 Supplementary Materials

Matlab/C-package: Matlab and C codes for the methods to create mean cloud images in the GEM system. The package also contains the image datasets used in the experiments presented in the article. (gem.tar.gz, unpacked by gunzip followed with tar -xvf)

Acknowledgments

The research of Jia Li was supported by the National Science Foundation under grants ECCS-1462230 and DMS-1521092. The research of Fuqing Zhang was supported by National Science Foundation under grant AGS-1305798. The authors would like to thank the reviewers and the editor for their very detailed and helpful comments and suggestions.

References

AGUEH, M. and CARLIER, G. (2011). Barycenters in the Wasserstein space. *SIAM J. Math. Analysis*. **43(2)** 904-924.

BANFIELD, J. D. and RAFTERY, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*. **49(3)** 803–821.

BENAMOU, J.-D., CARLIER, G., CUTURI, M., NENNA, L., and PEYRÉ, G. (2015). Iterative Bregman projections for regularized transportation problems. *SIAM J. Sci. Comp. (SJSC)*. **37(2)** A1111–A1138.

CARLIER, G., CHERNOZHUKOV, V., and GALICHON, A. (2016). Vector quantile regression: An optimal transport approach. *The Annals of Statistics*. **44(3)** 1165–1192.

CHEN, Y., YE, J., and LI, J. (2016). A distance for HMMs based on aggregated Wasserstein metric and state registration. *Proc. ECCV*. 451–466.

CUTURI, M. and DOUCET, A. (2014). Fast computation of Wasserstein barycenters. *Proc. Int. Conf. Machine Learning (ICML)* 685-693.

DON, P., EVANS, J. L., CHIAROMONTE, F., and KOWALESKI, A. M. (2016). Mixture-Based Path Clustering for Synthesis of ECMWF Ensemble Forecasts of Tropical Cyclone Evolution. *Mon. Wea. Rev.* **144** 3301-3320.

GIVENS, C. R. and SHORTT, R. M. (1984). A class of Wasserstein metrics for probability distributions. *Michigan Math Journal*. **31(2)** 231–240.

KALNAY, E. (2002). *Atmospheric Modeling, Data Assimilation and Predictability*. 1st Edition, Cambridge University Press.

LEE, H. and LI, J. (2012). Variable selection for clustering by separability based on ridgelines. *Journal of Computational and Graphical Statistics*. **21(2)** 315–337.

LEITH, C. E. (1974). Theoretical skill of Monte Carlo forecasts. *Mon. Wea. Rev.* **102** 409-418.

LI, J., RAY, S., and LINDSAY, B. G. (2007). A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*. **8(8)** 1687–1723.

Li, C., Srivastava, S., and Dunson, D. B. (2016). Simple, scalable and accurate posterior interval estimation. *Biometrika*. **104(3)** 665-680.

Li, J. and Wang, J. Z. (2008). Real-time computerized annotation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **30(6)** 985–1002.

Melhauser, C. and Zhang, F. (2012). Practical and intrinsic predictability of Severe and Convective Weather at the mesoscales. *Journal of the Atmospheric Sciences*. **69** 3350–3371.

Melnykov, V. and Maitra, R. (2010). Finite mixture models and model-based clustering. *Statistics Surveys*. **4** 80–116.

Meng, Z. and Zhang, F. (2008). Test of an ensemble Kalman filter for mesoscale and regional-scale data assimilation. Part IV: Comparison with 3DVar in a month-long experiment. *Mon. Wea. Rev.* **136** 3671-3682.

Minsker, S., Srivastava, S., Lin, L., and Dunson, D. B. (2014). Robust and scalable Bayes via a median of subset posterior measures. *arXiv preprint arXiv:1403.2660v3*.

Molteni, F., Buizza, R., Palmer, T. N., and Petroliagis, T. (1996). The ECMWF ensemble prediction system: Methodology and validation. *Quarterly J. Roy. Meteor. Soc.* **122** 73–119.

National Research Council; Division on Earth and Life Studies; Board on Atmospheric Sciences and Climate; Committee on Estimating and Communicating Uncertainty in Weather and Climate Forecasts (2006). *Completing the Forecast: Characterizing and Communicating Uncertainty for Better Decisions Using Weather and Climate Forecasts*. National Academies Press. *http://ftp.emc.ncep.noaa.gov/gc_wmb/yzhu/NRC_report/NRC.pdf*.

Rachev, S.-T. (1985). The Monge-Kantorovich mass transference problem and its stochastic applications. *Theory of Probability & Its Applications* **29(4)** 647–676.

Rabin, J., Peyré, G., Delon, J., Bernot, M. (2011). Wasserstein barycenter and its application to texture mixing. *Scale Space and Variational Methods in Computer Vision* 435-446.

Raftery, A. E., Gneiting, T., Balabdaoui, F., and Polakowski, M. (2005). Using Bayesian model averaging to calibrate forecast ensembles. *Mon. Wea. Rev.*. **133** 1155–1174.

Ray, S. and Pyne, S. (2012). A computational framework to emulate the human perspective in flow cytometric data analysis. *PloS one*. **7(5)** e35693.

Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Barker, D. M., Wang, W., and Powers, J. G. (2005). A description of the advanced research WRF version 2. *NCAR Tech. Note NCAR/TN-4681STR* 88 pp.

Sivillo, J. K., Ahlquist, J. E., and Toth, Z. (1997). An ensemble forecasting primer. *Weather and Forecasting*. **12(4)** 809–818.

Solomon, J., de Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., and Guibas, L. (2015). Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.* **34(4)** 66:1–11.

Sommerfeld, M. and Munk, A. (2017). Inference for empirical Wasserstein distances on finite spaces. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. **80(1)** 219–238.

Sorkine-Hornung, O. and Rabinovich, M. (2017). Least-square rigid motion using SVD. $https://igl.ethz.ch/projects/ARAP/svd\_rot.pdf$.

Srivastava, S., Li, C., and Dunson, D. B. (2015). Scalable Bayes via Barycenter in Wasserstein Space. *arXiv preprint arXiv:1508.05880v3*.

Toth, Z. and Kalnay, E. (1997). Ensemble forecasting at NCEP and the breeding method. *Mon. Wea. Rev.* **125** 3297-3319.

Ye, J., Wu, P., Wang, J. Z., and Li, J. (2017). Fast discrete distribution clustering using Wasserstein barycenter with sparse support. *IEEE Trans. on Signal Processing*. **65(9)** 2317–2332.