Visual Informatics 8 (2024) 67–79

Contents lists available at ScienceDirect

Visual Informatics

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

Research article

# The whole and its parts: Visualizing Gaussian mixture models

Joachim Giesen [a], Philipp Lucas [a], Linda Pfeiffer [b], Laines Schmalwasser [b], Kai Lawonn [a,*]

[a] Friedrich-Schiller-University Jena, Germany
[b] German Aerospace Center DLR-Institute of Data Science Jena, Germany

ARTICLE INFO

*Article history:*
Received 27 June 2023
Received in revised form 18 January 2024
Accepted 19 April 2024
Available online 12 May 2024

*Keywords:*
Gaussian mixture models

ABSTRACT

Gaussian mixture models are classical but still popular machine learning models. An appealing feature of Gaussian mixture models is their tractability, that is, they can be learned efficiently and exactly from data, and also support efficient exact inference queries like soft clustering data points. Only seemingly simple, Gaussian mixture models can be hard to understand. There are at least four aspects to understanding Gaussian mixture models, namely, understanding the whole distribution, its individual parts (mixture components), the relationships between the parts, and the interplay of the whole and its parts. In a structured literature review of applications of Gaussian mixture models, we found the need for supporting all four aspects. To identify candidate visualizations that effectively aid the user needs, we structure the available design space along three different representations of Gaussian mixture models, namely as functions, sets of parameters, and sampling processes. From the design space, we implemented three design concepts that visualize the overall distribution together with its components. Finally, we assessed the practical usefulness of the design concepts with respect to the different user needs in expert interviews and an insight-based user study.



## 1. Introduction

The work that we present here is motivated by our work on a visual analytics workflow for probabilistic programming (Klaus et al., 2023). In this workflow, we synthesize a mixed-type probabilistic program, that is, a program with categorical and continuous variables, from data. The probabilistic program specifies a multivariate, typically high-dimensional probability distribution. Users can iteratively change the distribution by modifying the program and visually observing the effects of the modifications. Since high-dimensional probability distributions cannot be visualized directly, we have adopted the well-established *slicing* approach for multivariate data visualization. In this approach, two operations, namely, *projections* and *selections*, are used to compute data slices that are smaller than the full data set and thus are easier to directly visualize. Projections restrict the data set to a subset of the variables, and selections only keep data points that satisfy a filter condition. Users can use the slicing operations to visually explore the space of smaller data slices with, typically, up to five or six dimensions. A data slice is essentially an attributed point cloud, where the continuous dimensions specify the coordinates of the points and the categorical dimensions attach attributes to the points. In the exploratory slicing approach, that was first implemented in some generality within the Polaris project (Stolte et al., 2002a,b), the user provides the filter condition and specifies the variables of interest and how they should be mapped to visual marks, such as points, lines, and areas, and channels that specify the marks' appearance. The continuous part of a data slice is mostly mapped to points, as in scatter plots (Friendly and Denis, 2005) and scatter plot matrices (Hartigan, 1975), or lines, as in parallel coordinate plots (Inselberg and Dimsdale, 1990). The categorical part is mapped to identity channels such as color or shape.

The key insight that makes it possible to adopt the slicing approach also for multivariate probability distributions is a non-trivial correspondence between the projection and selection operations on multivariate data and the *marginalization* and *conditioning* of a multivariate probability distribution. The key component of a probabilistic programming language is its inference engine that implements the marginalization and conditioning operations. By marginalization and conditioning, a multivariate probability distribution can be sliced into smaller probability distributions that can be directly visualized. As in multivariate data visualization, users can thus use the operations of marginalization and conditioning to explore the space of smaller slices of multivariate probability distributions. Furthermore, the direct correspondence of the slicing operations on data and models makes it possible to visualize model and data slices together in one visualization. Therefore, users can visually explore probabilistic models and changes to these models by exploring model slices together with the corresponding data slices. The visualization of

* Corresponding author.
*E-mail address:* kai.lawonn@uni-jena.de (K. Lawonn).

https://doi.org/10.1016/j.visinf.2024.04.005

data slices, that is, the mapping of data dimensions to visual marks and channels, is a mainstay of visualization research, whereas the visualization of model slices has not received much attention yet.

Here, we focus on the visualization of a specific type of model slice, namely mixture slices, where a subset of the categorical variables have been marginalized out. Mixture slices are important for the evaluation of probabilistic programs, because the inferential power of probabilistic programs is exactly in the interactions among the variables. Mixture slices bring these interactions, especially interactions between categorical and continuous variables, to the forefront. To keep the discussion focused, we restrict ourselves to a specific mixture model, namely a mixture of Gaussians, which is the margin of a CG distribution (Olkin and Tate, 1961; Dempster, 1973), where the single categorical variable has been marginalized out.

Gaussian mixture models are interesting in their own right. They are a staple of machine learning, see for example (Bishop, 2007, Chapter 9), and are still in use, even in state-of-the-art deep learning pipelines (Yang et al., 2019). A Gaussian mixture model with $k$ components is formally given by the following probability density function on $\mathbb{R}^d$,

$$p(x) = \sum_{i=1}^{k} w_i p_i(x) \quad \text{with} \quad p_i(x) \sim \mathcal{N}(\mu_i, \Sigma_i)$$

and non-negative weights $w_i$ that sum up to one. That is, the whole density function $p(x)$ is a weighted sum of its parts (components) $p_i(x)$. Here, the $\mathcal{N}(\mu_i, \Sigma_i)$ are $d$-variate Gaussians with mean vectors $\mu_i$ and covariance matrices $\Sigma_i$.

In this work, we investigate effective visual encodings that support the understanding of multivariate Gaussian mixture models. Here, understanding has four aspects, namely, understanding the whole density, understanding the different components of the density, understanding interactions among the components, and understanding the contributions of the components to the whole distribution. These four aspects constitute four generic needs that guide our design process of visual encodings of Gaussian mixture models for supporting a qualitative understanding of the encoded probability density functions, namely, the whole as well as its parts. In a literature review of articles that use Gaussian mixture models in application domains such as astronomy, epidemiology, or environmental science, we identified examples of domain-specific tasks that can be mapped to the four domain-independent user needs, proving that all four needs are indeed relevant in applications. To meet the needs, the challenge from a visualization perspective is to represent all the components in such a way that it is clear how they relate to each other and how they combine in a non-trivial way to form a whole.

We are not aware of previous work in visualization that has addressed this problem, except for the much simpler problem of visualizing proportions, for instance, by stacked bar charts (Talbot et al., 2014), where simple components trivially combine into the whole.

Before discussing our contributions, we would like to reiterate that our motivation is the validation and improvement of probabilistic programs by visually comparing data and model slices. Since the visualization of multivariate data slices is well understood, we focus on the visualization of model slices, here, on the example of Gaussian mixture model slices. Specifically, our model slice visualizations support asking parts-and-the-whole type questions about the models, but do not support data tasks like clustering or outlier detection. Data tasks are supported by interactive visualizations of data slices. Furthermore, the model slice visualizations alone do not support model validation tasks like estimating a good number of components. However, as we have pointed out already, visualizing model slices is an integral component of the slicing-based approach for probabilistic model visualization. Here, our specific contributions are:

- A first systematic study of a non-trivial parts-and-the-whole type problem from a visualization point of view. Here, non-trivial means that the whole is more than the sum of its parts, for instance, in the sense that the number of modes of the mixture can be very different from the number of its components.
- A structuring of the design space of visual encodings for Gaussian mixture models (Section 3).
- A discussion and evaluation by expert feedback on three fairly different design concepts from the design space (Sections 4 to 6).
- A more extensive evaluation of the three design concepts in an insight-based user study with non-expert users (Section 7).

## 2. User needs analysis

In principle, applications and tasks on Gaussian mixture models can relate either to understanding the whole density (N1), the different components (N2) and their relationships (N3), or the relationship of the components to the whole density (N4). It is, however, not directly obvious that all four needs actually arise in practice. Therefore, we actively searched the literature on applications of Gaussian mixture models for practical examples of tasks that relate to the four needs. That we were able to find such examples is proof that all four needs are indeed practical. We found tasks that relate to the four needs by a two-stage literature search. In the first stage, we identified research articles about applications of Gaussian mixture models that involve a need for visualization. In the second stage, we extracted application-specific tasks from the identified articles, which we then associated with the four domain-independent user needs.

### 2.1. Identification of relevant articles

The body of work on Gaussian mixture models is vast, as exemplified by the more than 1.3 million hits on Google Scholar for the search phrase *"application of Gaussian mixture model"*. An exhaustive manual analysis of these hits is infeasible. Therefore, we resorted to a limited, greedy strategy of searching and filtering. We set an upfront *limit* of 100 articles to be considered for our analysis and selected these 100 articles as follows: We searched on Google Scholar with the search phrase *"application of Gaussian mixture model"* and took the top-ranked articles as starting points for a subsequent *greedy* search. In the greedy search, we followed relevant hints like citations and keywords from the top articles to refine our search on Google Scholar. This way we selected 103 articles, slightly more than our limit, such that we cover a broad range of applications. For these articles, we checked whether tasks related to needs N1–N4 were indicated in the abstracts. We only considered articles further in which such tasks were indicated, which reduced the number to 42. We then scanned the full texts of these remaining articles and again only kept the relevant ones. Thereby, we finally found nine articles (Mathur et al., 2018; Liu et al., 2012; Lee et al., 2012; Allili et al., 2007; Kawabata, 2008; Aubert et al., 2016; Li et al., 2016; Wang et al., 2010; Shin et al., 2009) from which we were able to derive specific tasks that related to the abstract needs N1–N4. All nine articles also describe a need for understanding Gaussian mixture models, the need that was at least partially addressed by visualizations.

Articles that have been filtered out during the last two steps typically utilize Gaussian mixture models in some automatic machine learning pipelines. A good example is the work by Bernaille et al. (2006) that uses mixture models for the automatic association of software applications with network traffic. In such a work, the goal is not data-driven knowledge discovery. Instead, the model serves a technical means, typically clustering or classification, and is evaluated by simple performance scores. Therefore, no attempt is made to understand the structure of the mixture model. Hence, there is also no need to visualize the model.

### 2.2. Application-specific tasks and domain-independent needs

In the second stage of our literature analysis, we first identified application specific tasks in each of the nine relevant articles, before we associated them to the four generic domain-independent user needs N1–N4 in two discussion sessions among the authors. We consider the associated tasks as a proof of relevancy for the corresponding needs.

**N1 (Understanding single components).** Users want to understand individual components. This involves identifying a component, assessing its location, its spatial extent, and also its orientation. This need arises in seven out of nine articles (Mathur et al., 2018; Lee et al., 2012; Allili et al., 2007; Aubert et al., 2016; Li et al., 2016; Wang et al., 2010; Shin et al., 2009). For example, Lee et al. (2012) who model and classify pulsar distributions, notice that *"the ellipse for high magnetic field pulsars is quite extended"*. And Li et al. (2016) who are interested in anomaly detection in flight operations, recognize *"ten of the 35 mixture components [...] as flight operations that are well known during approach phase"*.

**N2 (Relating components).** Users want to relate the components of a model to each other. This involves understanding their relative location, shape and orientation, as well as their degree of overlap, the actual location of overlap, and also the relative weight. We identified this need in the same seven articles that also exhibit need N1. It arises mostly from the task of understanding the differences between components in order to find a domain-specific explanation for the existence of multiple components. For example, Lee et al. (2012) notice the different orientations of principal axes of two components, leading them to the hypothesis that *"the MSPs (millisecond pulsars) have two different origins or evolutionary tracks"*. As another example, Li et al. (2016) note that the *"touchdown [component] has significantly more observations than other clusters"*.

**N3 (Understanding the whole distribution).** Users also want to assess the whole distribution. This involves an understanding of its shape, its extent, and the identification of modes (local maxima). Six out of the nine articles feature this need (Mathur et al., 2018; Liu et al., 2012; Lee et al., 2012; Allili et al., 2007; Kawabata, 2008; Aubert et al., 2016). For example, Mathur et al. (2018) who model the spread of dengue fever observe that while individual components are of interest since they can be interpreted as *"local hot spots"* the total spread can be used to *"generate early warning for dengue incidences"* to control the virus.

**N4 (Relating components to the whole distribution).** Finally, users want to relate the components to the whole distribution, mostly because they need to understand how the whole is composed of its parts at different locations. Three out of nine articles express this need (Lee et al., 2012; Allili et al., 2007; Aubert et al., 2016). For example, Lee et al. (2012) want to understand how four components jointly form and *"explain"* the distribution of so called normal pulsars. Similarly, Aubert et al. (2016) who infer general hydrochemical rules from large environmental data sets understand *"the overall nitrate concentration as a composition of three components"*.

In Section 6 we will relate the abstract user needs N1–N4 also to domain-independent, actionable tasks that we elicited from experts, who are working with Gaussian mixture models in various contexts and applications. Finally, we would like to mention another frequent task in the literature, namely, the assignment of semantics to the components of a mixture model. Analysts want to understand what a component represents and what it means. For example, Lee et al. (2012) assign semantics like *"high magnetic field pulsars"* or *"old pulsars close to death"* to individual components of the mixture, and Allili et al. (2007) identify one component as representing *"noise in the data"*. Important as such a task is, we cannot consider it a generic user need because it fundamentally depends on the application domain.

## 3. Representing Gaussian mixture models

Gaussian mixture models have three very distinct, but individually complete, representations that make the design space for their visualization interesting: First, $d$-dimensional Gaussian mixture models are $d$-variate functions. Second, these functions are completely specified by a finite set of parameters, namely, the mean vectors and the covariance matrices of the components, and their mixture coefficients. Third, Gaussian mixture models define a hierarchical sampling processes, where we first sample a component before we sample from the component. All three representations can be used for visualizing the whole Gaussian mixture model as well as its components. However, there is no need to use the same representation for both, the whole and its parts. Hence, the choice of representation for the whole and for its parts is a relevant design decision.

In the following we describe the three representations and discuss possible visual encodings, that is, mappings to visual marks and channels. We start with the representation by a sampling process. The sampling representation, however, is not well suited for our overall goal of slicing-based model validation, where we compare the model and corresponding model slices. A sampling process represents the model, essentially, by another data set, which means that the large body of work on visualizing multiclass data sets can be reused for the sampling-based model visualization. Then, however, data and model visualization are similar and difficult to distinguish. Furthermore, the sampling process representation is the least accurate among the three representations, because it is, by definition, only an approximation. Therefore, after the brief discussion below, we will not consider the sampling process representation further in the remainder of this article.

### 3.1. Sampling process representation

We can draw samples from a Gaussian mixture model in a two-stage sampling process: First, we draw one class label out of the $k$ labels for the components, where the class probabilities are the weights $w_i$. Second, we draw a sample point from the Gaussian that corresponds to the class label from the first stage. For any given Gaussian mixture model, we thus can generate multiclass data sets of arbitrary size.

The visualization of multiclass data sets has been widely studied. Sarikaya and Gleicher (2018) provide a thorough discussion of the design space of multiclass scatter plots. Their paper discusses common design challenges like visual complexity (too many points) that are also relevant in our context. Here, we can control the total number of sample points, however, it is not straightforward to define a perceptually good number as can be seen in Fig. 1.

Multidimensional multiclass data sets can also be visualized by parallel coordinate plots or clustered parallel coordinate plots that aggregate the lines for the different classes. Similarly, as for multiclass scatter plots, color and hue preserving color blending

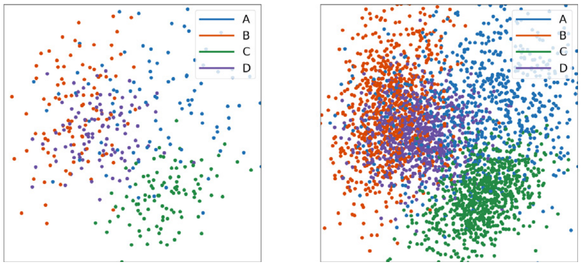


**Fig. 1.** Two sample visualizations with different numbers of sample points. Together, all the sample points represent the whole mixture distribution, whereas sample points of the same color represent the component. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

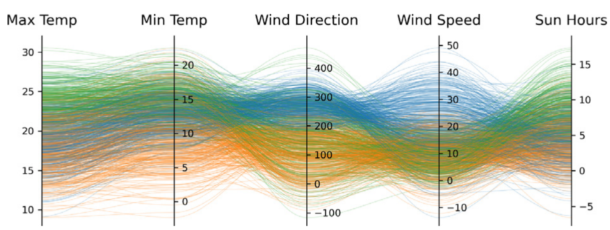


**Fig. 2.** A parallel coordinates plot that shows 1200 sample points drawn from a five-dimensional Gaussian mixture model with three components derived from the weather data set (The National Climate Database, 2021).

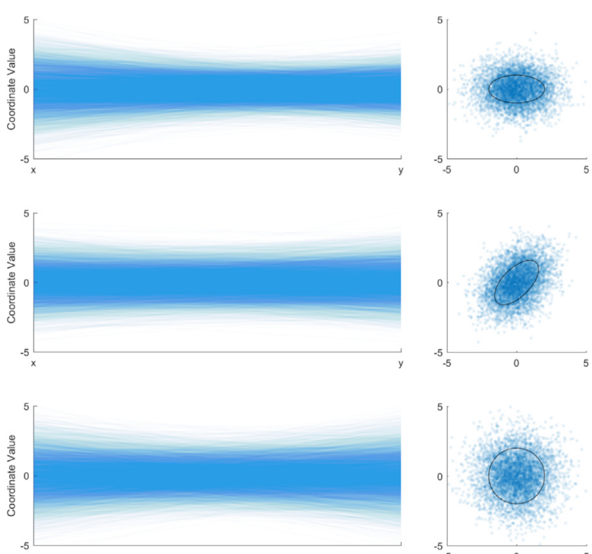


**Fig. 3.** The parallel coordinate plots and the scatter plots of three Gaussians are shown. The first and second Gaussian differ by a 45° rotation.

can be used to distinguish the different classes. An example is shown in Fig. 2

However, parallel coordinate plots are not well suited to convey the spatial extent and orientation of sampled Gaussians. To illustrate these problems of parallel coordinates for visualizing Gaussians, we defined the following three two-dimensional Gaussian distributions: We set the mean to (0, 0) for all Gaussians and set the covariance matrices to:

$$\Sigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}, \quad \Sigma_3 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

Note, that $\Sigma_2$ results from a counter-clockwise 45° rotation of the first Gaussian. From the individual Gaussians, we sampled 5,000 points and visualized them as shown in Fig. 3 by parallel coordinates plots and also by scatter plots. Neither the rotation nor the difference in spatial extent between the second and the third Gaussian are clearly recognizable from the parallel coordinate plots. That is, parallel coordinate plots can be helpful for showing data points, but cannot help to understand the spatial extent and orientation of the Gaussians, which is important for needs N1 and N3.

### 3.2. Function representation

The most direct representation of a Gaussian mixture model is as a function. The density of a Gaussian mixture model is a positive function $p : \mathbb{R}^d \to (0, \infty)$, and so are all of its components $p_i : \mathbb{R}^d \to (0, \infty)$, that is, the mixture and its components are scalar height fields.

Multivariate Gaussian mixture models have a particularly nice property, namely, marginal distributions of these models are just lower-dimensional Gaussian mixture models. This property allows us to decompose a high-dimensional mixture model into two-dimensional marginals that can be visualized directly. That is, we can directly adopt the slicing approach also for high-dimensional Gaussian mixture models by visualizing two-dimensional slices, exactly as in scatter plot matrices for high-dimensional data. In the function representation, two-dimensional Gaussian mixture models are scalar fields over the Euclidean plane. Such scalar fields are typically visualized by contour plots or extensions thereof Mayorga and Gleicher (2013). Different fields can be distinguished by color. We discuss a design concept for the function representation using contour plots and color in the next section. Another straightforward extension of contour plots to several simultaneous scalar fields is a juxtaposition of contour plots. In our case, we could use one contour plot for the whole density function and one for each component. However, due to the missing alignment of (at least one of) the axes in the different contour plots, it becomes harder to compare them intuitively and quickly. Also, multiple plots occupy more screen space.

The ideas behind parallel coordinate plots are not directly applicable to the function representation, because there is no direct mapping from density values to coordinate lines.

### 3.3. Parameter representation

Finally, a Gaussian mixture model is also fully determined by a set of parameter triples,

$$\{(w_i, \mu_i, \Sigma_i) : i = 1, \ldots, k\},$$

see Section 1. Location parameters on screen space can be derived from projections of the mean vectors $\mu_i \in \mathbb{R}^d$, and the positive definite covariance matrices $\Sigma_i$ themselves can be represented by their orthogonal eigenvectors (directions) and positive eigenvalues (scales). In Section 4 we derive a glyph and thus a design concept from the latter representation of the covariance matrices. Using symbols/glyphs on top of a map is a fairly common technique for displaying additional attributes. For instance, Brewer and Campbell (1998) use superimposed pie charts for visualizing multivariate data on a map. We pursue this idea further in Section 4.

Ideas from parallel coordinate plots could be adapted for use with the parameter representation. The mean vectors $\mu_i$ can be mapped directly to coordinate lines, and from the covariance matrices we can compute $d$ principal axes providing $2d$ more points that can be mapped to coordinate lines. However, these plots suffer from the same limitations as the plots in Fig. 3.

**Table 1**
Visualization techniques and representations used in the design concepts.

| | | Contours | | Principal axes | | Pie charts | |
|---|---|---|---|---|---|---|---|
| | | *Parts* | *Whole* | *Parts* | *Whole* | *Parts* | *Whole* |
| Visual encoding | Color blending | ✓ | | ✓ | | | |
| | Glyphs | | | ✓ | | ✓ | ✓ |
| | Contour plots | ✓ | ✓ | ✓ | ✓ | | |
| Representation | Function | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Parameter | | | ✓ | | | |

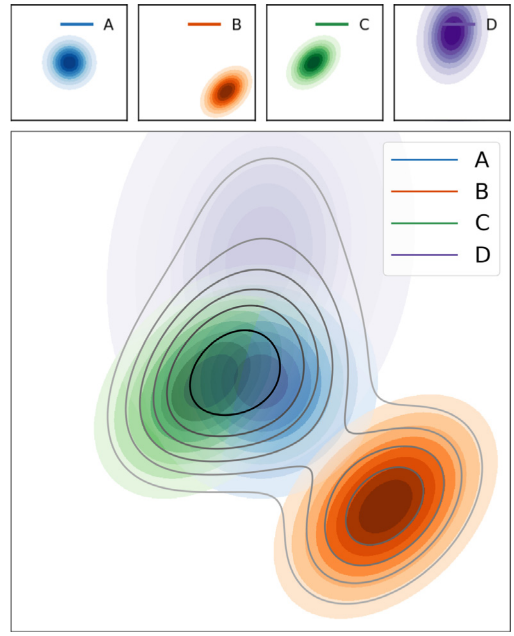


**Fig. 4.** Contours concept. On the top: Juxtaposition of contour plots for the individual not-weighted components. At the bottom: Contour plots for the weighted components and overlaid contour plot for the whole density.

## 4. Design concepts

The discussion in the previous section shows that the design space for the visualization of Gaussian mixture model is structured but vast and thus cannot be explored exhaustively. Therefore, we focus here only on a special but rich class of visual representations, namely, the generalization of the slicing approach to Gaussian mixture models. As noted before, this generalization becomes possible because marginal distributions of Gaussian mixture models are again Gaussian mixture models. At the core of the slicing approach are visualizations of two-dimensional slices, that is, two-dimensional marginals. Nevertheless, even in the two-dimensional case, the challenge to visually represent both the whole and its parts remains. In the following, we describe three specific design concepts for two-dimensional Gaussian mixture models and discuss how they relate to the four user needs (N1–N4) from Section 2. The concepts, namely, the *contours*, the *principal axes*, and the *pie charts*, can be compared in the teaser on a simple Gaussian mixture model and in Fig. 9 on a more complex model. In Section 5 we show how to scale these concepts to higher dimensions by using the operation of marginalization. In this section we also discuss that the slicing approach compares favorably to parallel coordinate plots that directly generalize to higher dimensions.

The three design concepts sample the design space along three dimensions of independent design decisions, namely *representation* (parameters, function, or process), *object of visualization* (component or whole), and *visual encoding*. Table 1 summarizes how the concepts relate to these design choice dimensions. In Section 4.4, we provide more information about design decisions common to all three design concepts, including a detailed description of our color blending procedure.

### 4.1. Contours concept

The contours concept directly visualizes the density functions, that is, the function representations of the whole and the parts by contour plots. Each component is visualized by a filled contour plot, where the color of the area between two adjacent isolines encodes the average density of the component in that area. The contour plots of the components are combined by blending the colors of corresponding (overlaying) pixels, using the average densities of the components' contour area at this pixel (location) as weights. The whole density is visualized by a contour line plot that is superimposed on the visualization of the components. The contour lines are colored by mapping isovalues to gray scale values. Fig. 4 shows an example.

*Discussion of the contours concept*

A common approach to visualizing Gaussian mixture models is superimposing independent visualizations of the components by contour plots, that is, superimposing individual visualizations for every Gaussian from the mixture. The contours concept improves upon this naive approach in several ways. It adds a direct visualization of the whole density, that is missing in the naive approach. By following Type-J multiplexing (Chen et al., 2014), it distinguishes clearly between the components (filled contour plots) and the whole density (contour line plots). The overlapping contour lines for the components in the naive approach easily lead to visual clutter and make it hard to assess local composition. Instead, the contours concept employs color mixing to eliminate visual clutter and uses common scales to assure the effective comparability of components.

Contour plots allow for a detailed understanding of the densities' extent in the two quantitative dimensions, and for an assessment of density values at any point in the visualized region of interest of the mixture. Both are important for the understanding of the parts (N1) and the whole (N3), as well as for the spatial relation between components (N2). However, the assessment of density values is affected by the choice of the threshold below which a component's density function value is mapped to the background color. Here, we decided to cut off each component individually instead of using a global threshold. This ensures an equitable representation of the individual components and helps to understand their individual distributions (N1). A global threshold, instead, would better support the comparison of the components' density values in regions of low density (N2). The chosen color blending method of single-hue color schemes, see Section 4.4 for technical details, helps to identify at any point the component that contributes most to the whole distribution (N4).

Altogether, we expect this design to lead to more insights relevant to user needs N1 (single components) and N3 (whole distribution), since their spatial characteristics are clearly visible. N1 may be slightly better supported than N3, since the use of colored regions is more salient than using contour lines.

### 4.2. Principal axes concept

This concept combines a contour plot for the whole density (like in the contours concept) with one colored, cross-shaped glyph for each component. The glyphs are derived from the parameter representation of the components and provide more compact visualizations of their shape. They are constructed in four steps; see also Fig. 5.

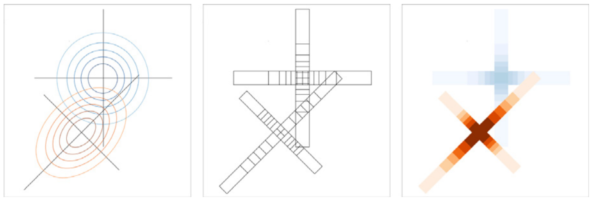

**Fig. 5.** Construction of the glyphs for the principal axes concept. Left: Intersection of contour lines with the principal axes of two components centered at their means. Middle: Rectangular subdivision of the strips that are computed from the intersection of the principal axes with the contour lines. Right: Final representation after color blending. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

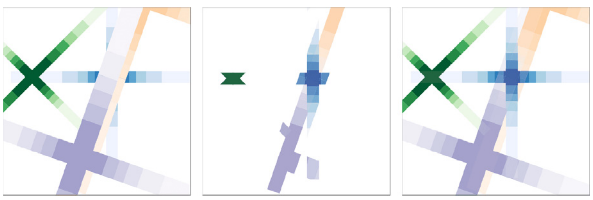

**Fig. 6.** Effect of color blending on the iteratively computed intersections of contour segments. Left: No blending (depth ordered). Middle: color blending for intersecting regions only. Right: final result. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

First, the shape of the glyph is computed for each component. We use the fact that for any positive definite $(2 \times 2)$-matrix $\Sigma^{-1}$ (inverse covariance matrix) the solution to the equation $x^\top \Sigma^{-1} x = 1$ is an ellipse centered at the origin. Let $\lambda_1 > \lambda_2 > 0$ be the eigenvalues of $\Sigma^{-1}$ and $e_1$ and $e_2$ be the corresponding normalized eigenvectors, then $\frac{e_1}{\sqrt{\lambda_1}}$ is the minor axis and $\frac{e_2}{\sqrt{\lambda_2}}$ the major axis of the ellipse. We thicken the line segments $(\mu \pm e_1)/\sqrt{\lambda_1}$ and $(\mu \pm e_1)/\sqrt{\lambda_2}$ into strips, where the width of both strips is set to $\frac{1}{10} \cdot \frac{e_1}{\sqrt{\lambda_1}}$. Thus, the thickness of the glyphs is proportional to the spatial extent of the respective Gaussians, that is, for peaked Gaussians the glyphs are less thick than for non-peaked Gaussians since the spatial extent is larger for non-peaked Gaussians than for peaked Gaussians. The strips define the glyph's shape. Second, we subdivide the strips into rectangles (*contour segments*) at the intersection points of the contour lines for each component. To each contour segment, we can associate the corresponding component and its assigned color (hue), as well as the isovalue of the corresponding contour line. Third, we compute the arrangement of all contour segments, that is, the arrangement of polygons that result from intersecting all the contour segments. For each polygon in the arrangement, we know all contour segments that cover it, that is, have contributed to the intersection. Therefore, we can color each polygon by blending the colors from all covering contour segments using the associated isovalue and the components' weights. The importance of color blending on the intersections of the glyphs' strips is demonstrated in Fig. 6. Finally, a gray scale contour plot for the overall density is overlaid on the combined glyphs.

*Discussion of the principal axes concept*

Compared to the contours concept, the use of glyphs for visualizing the components reduces visual clutter. Thus, the overall distribution is more salient, which could lead to more insights relevant to user need N3 (whole distribution). Moreover, it simplifies the augmentation of the visualization by additional layers of information. Furthermore, the parameter representation eliminates the need for a threshold value below which density values are mapped to the background color, since the lengths of the glyph's axes are finite. Also, the explicit encoding of the components' means and orientations by the glyphs' axes can improve understanding of components (N1) and their relations (N2). However, the cognitive load for the user might increase, because of the necessary mental expansion of glyphs to the shapes of their components, which could impair the understanding of relations (N2 Relating components, N4 Relating components to the whole distribution).

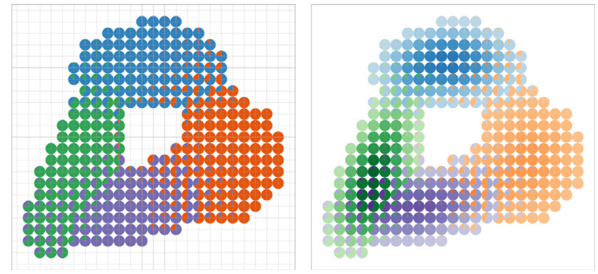

**Fig. 7.** Construction of the pie charts concept. Left: Pie chart glyphs for the components at grid points. Right: Using the lightness channel for encoding the whole density function.

**Fig. 8.** Alternative visual channels for the whole density function. Left: Size. Right: Size and lightness.

### 4.3. Pie charts concept

Assessing the proportions of the components' densities at a single point is difficult with the contours and principal axes concepts, because they use color blending that aggregates the contributions from the different components. This motivated us to directly encode the proportions visually. For that, we discretize the region of interest into a grid and encode the proportions of the components at the grid points using pie chart glyphs. Each glyph encodes the relative density of the components at its location (grid point) by the relative size of its circular segments. Note that no color blending is required, as components are represented by their assigned color (hue). The density of the whole Gaussian mixture model is then visualized by mapping its density value at a grid point to either a lightness or saturation channel for the glyph's color coding. Alternatively, the density values can also be mapped to the size of the pie chart glyphs. The construction of the pie charts concept is illustrated in Fig. 7 and alternative visual channels for the whole density function are shown in Fig. 8.

*Discussion of the pie charts concept*

By construction, the pie charts concept should allow for a better comparison of the density values (N2 Relate components) and the contribution of components to the whole distribution (N4 Relate components to the whole distribution). While the discretization might make it more difficult to perceive the shape of a distribution (N1 Single components, N3 Whole distribution), the

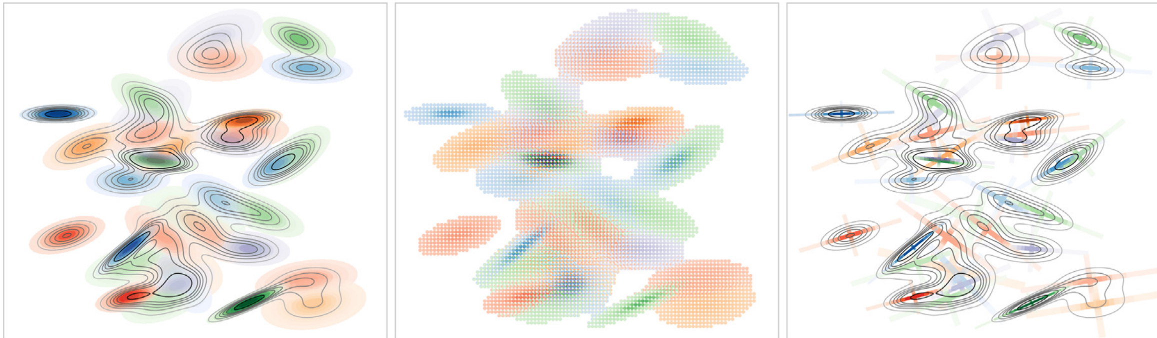

**Fig. 9.** Comparison of all three design concepts (contours, pie charts, and principal axes) on an artificial Gaussian mixture model with 35 components. All means, variances, and weights were drawn at random. The weights are drawn from a Dirichlet distribution with a large concentration parameter, resulting in a small variance of the weights.

Gestalt principles of closure appear to provide a remedy at least to some degree. Shape understanding with the pie charts concept can benefit from interactively adjusting the level of discretization.

### *4.4. Common design decisions*

In this section we discuss common design decisions regarding the region of interest, color, color scheme quantization, and the choice of isovalues that we use for all three design concepts.

Since mixtures of Gaussians are probability density functions with unbounded support we need to define a *region of interest*, that is, a rectangular subset of the Euclidean plane, on which we visualize the mixture model. The region of interest should cover the interesting regions of the whole distribution as well as its parts. By default, we choose the region of interest as the axes-parallel rectangle such that for all components the minimal distance of the means to the boundary of the rectangle is at least $4 \cdot \sqrt{cov}$ where $cov$ is the covariance along an axis. For the pie charts concept, the region of interest is covered by regular grids.

All three concepts map the data's two quantitative dimensions on the spatial x- and y-channels. Since hue is the most efficient channel for encoding categorical variables except for spatial separation (Munzner, 2015), we decided to differentiate between the components by using different hues. In addition to the categorical information about the components, we also want to map quantitative information about density values to color. This mapping problem boils down to choosing a color scheme for each component and to quantizing continuous density values.

**Color schemes.** For the choice of color schemes we have three requirements, namely, (1) colors assigned to a single component should be sequentially ordered such that the naturally ordered density values can be represented accordingly, (2) colors assigned to different components should be clearly distinguishable, and (3) blending of colors for different components should not create colors that are close to colors that are assigned to other components. These requirements can be satisfied by monochromatic color schemes, where a single hue is modified by changing its lightness. For single-hue schemes, additionally, the saturation can be modified. Therefore, we chose to use single-hue schemes from the ColorBrewer Website (Brewer, 2013; Harrower and Brewer, 2003).

**Quantization of color schemes.** Quantization means to choose discrete colors from the respective color schemes to encode the overall density as well as the densities of the components. We have two requirements for the quantization: First, we want to use the same number $n$ of colors for every component of the Gaussian mixture model, and second, we want to make density values for different components comparable. To achieve this, we choose color schemes that adapt to the local scales as well as to the global scale. The local scale for the $i$th component is given by its maximal density value $p^i_{\max} = \sup\{p_i(x) : x \in \mathbb{R}^2\}$, and the global scale is given by the overall maximum density value $p_{\max} = \max_i\{p^i_{\max}\}$. Fig. 10 shows an example for a set of quantized color schemes that are locally as well as globally adapted.

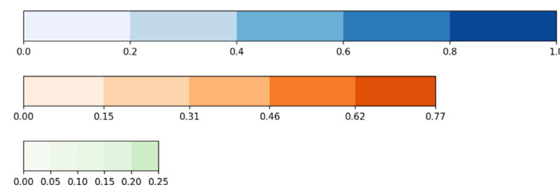


**Fig. 10.** Adaptive color schemes for three components and quantization level $n = 5$. Here we have $p_{\max} = p^1_{\max} = 1$ (blue component), $p^2_{\max} = 0.77$ (red component), and $p^3_{\max} = 0.25$ (green component). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Threshold.** In practice a small problem remains, namely, color schemes that are quantized as described above never map to the white (background) color, because the density function of a Gaussian and thus a Gaussian mixture model always evaluates to values larger than zero. Hence, we decided to introduce a threshold of $p^i_{\max} \cdot \varepsilon$ for some small $\varepsilon > 0$ and map density values that are smaller than $p^i_{\max} \cdot \varepsilon$ to the background color. Such a threshold is, however, not necessary for the principal axes concept, see Section 4.2.

**Color blending.** Color blending is done in some color space, where the blending operation can be reduced to some arithmetic operation in the color space. Here, we chose to work in the perceptually uniform CIE L*a*b color space. Typical blending operators, like alpha blending or the Porter-Duff-Source-Over operator (Porter and Duff, 1984), take a set of colors $c_1, \ldots, c_n$ (vectors in color space) that are depth ordered from back to front together with weights (opacities) $\alpha_1, \ldots, \alpha_n \in [0, 1]$, and recursively compute the blended color as a linear combination of the given colors. The coefficients in the linear combination are typically non-linear functions of the opacities. For instance, the formula for basic alpha blending is given as

$$c = \sum_{i=1}^{n} \Big( \prod_{j=i+1}^{n} (1 - \alpha_j) \Big) \cdot \alpha_i c_i,$$

which can be derived from the recursive scheme

$$\hat{c}_1 = \alpha_1 c_1 \ \text{ and } \ \hat{c}_i = (1 - \alpha_i)\hat{c}_{i-1} + \alpha_i c_i \ \text{ for } \ n \geq 2.$$

The blended color is then $c = \hat{c}_n$. We tried several modifications of this simple scheme and tested them in different scenarios for our design concepts. Finally, we settled on the following recursive schemes for the coefficients

$$\hat{\alpha}_2 = \frac{\alpha_1}{\alpha_1 + \alpha_2} \text{ and } \hat{\alpha}_i = \frac{\hat{\alpha}_{i-1}}{\hat{\alpha}_{i-1} + \alpha_i} \text{ for } n \geq 3,$$

and colors

$$\hat{c}_i = \hat{\alpha}_i \hat{c}_{i-1} + (1 - \hat{\alpha}_i) c_i \text{ for } n \geq 2.$$

Again, the blended color is $c = \hat{c}_n$.

**Choosing isovalues as quantiles of the density function.** A contour plot is a collection of contour lines that is determined by the choice of isovalues. A contour line for a given function and isovalue is the set of points at which the function takes the isovalue. For scalar fields with a physical interpretation, isovalues are often chosen such that the corresponding contour plot highlights geometric/topological information about the scalar field, namely its critical points. Here, it is more important to provide a perceptual uniform representation of probability density functions than to highlight their topological features. Therefore, we decided to choose the isovalues uniformly. This still leaves two options, namely, either choosing the isovalues uniformly in the range of the density function, or corresponding to uniformly spaced sublevel sets (percentiles) of the density function. In the latter option, the integral over the areas between two consecutive isolines always gives the same probability. In our case, the difference between the two options is usually not visually significant. Hence, we decided to use percentiles, because of their well-defined semantics in terms of probability.

### 4.5. Resolution parameters of the design concepts

All three design concepts are parameterized by one or more resolution parameters that we summarize below.

**Contours concept.** 1. Number of isovalues: Always, the same number of isovalues is used for all the components and for the whole density function. 2. Sampling rate of the density function along the two spatial dimensions in the region of interest.

**Principal axes concept.** 1. Number of isovalues: Always, the same number of isovalues is for the all components and for the whole density function. 2. Sampling rate of the density function along the two spatial dimensions in the region of interest. 3. Thickness of the principal axes strips: The thickness of a principal axis strip is always set proportional to $1/\sqrt{\lambda}$, where $\lambda$ is the eigenvalue corresponding to the principal axis. As a default value for the proportionality factor, we have used the value $\frac{1}{10}$.

**Pie charts concept.** 1. Grid size: The region of interest is covered by a regular square grid. The density functions of the components and the whole density function are sampled at the grid's vertices.

## 5. Multivariate Gaussian mixture models

So far, we have only discussed the visualization of two-dimensional (bivariate) Gaussian mixture models. Here, we argue that the visualization of high-dimensional mixture models can be reduced to the two-dimensional case by standard techniques from multidimensional data visualization. The probably best known example of such a technique is a scatter plot matrix (SPLOM) (Hartigan, 1975) that facilitates the analysis of high-dimensional quantitative data by projections onto two-dimensional coordinate planes. Mixtures of Gaussians also have natural projections onto two-dimensional coordinate planes, namely their two-dimensional marginals. These marginals can be derived analytically and thus be computed efficiently. Furthermore, the marginals are again Gaussian mixture models. Hence, two-dimensional marginals can play exactly the same role in visualizing Gaussian

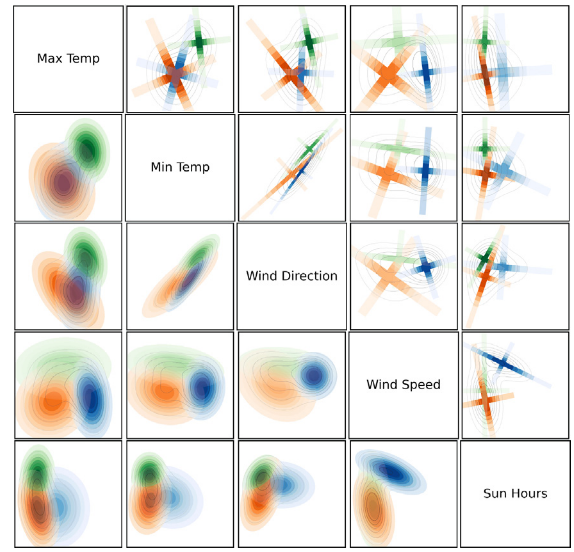


**Fig. 11.** A 5 × 5 matrix of all two-dimensional marginals of a five-dimensional Gaussian mixture model with three components derived from the weather data set (The National Climate Database, 2021) visualized using the contours (lower triangular matrix) and principal axes concept (upper triangular matrix), respectively. The model and the colors assigned to the mixture components are aligned with Fig. 2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

mixture models that projections onto two-dimensional coordinate planes play for scatter plots and scatter plot matrices. An example of a marginal plot matrix is shown in Fig. 11.

Like in SPLOMs, each two-dimensional marginal occurs twice in the matrix layout. Here we can make use of this redundancy and show two complementary design concepts at once, one in the lower and one in the upper triangular matrix, respectively. If the number of dimensions becomes too large to visualize all pairs, then adapting techniques from the data domain like scagnostics (Wilkinson et al., 2005) for choosing informative dimension pairs could be an interesting option and avenue for future research.

Note, that for general probability distributions, it is often not possible to analytically derive marginals or even approximate them efficiently. Even worse, for more general multivariate functions than probability distributions, no general projection technique like marginalization is known. Hence, completely different techniques for their visual exploration have been developed, like, for instance, the Sliceplorer technique (Torsney-Weir et al., 2017).

Parallel coordinates plots naturally scale to higher dimensions. An example of the same data as for the marginal plot matrix above is shown in Fig. 2 in Section 3.1. While some spatial separation between the different components of the mixture model is visible in the parallel coordinate plots, the separation can be seen more clearly in the marginal plot matrix. Also, the separation can be interpreted more directly in the marginal plot matrix. Furthermore, it is more difficult to estimate density values from the parallel coordinates plot. The latter problem is likely to be exacerbated when edge bundling is used.

## 6. Evaluating the design concepts

In this section, we report on semi-structured interviews that we have conducted with five scientists and data analysts who have worked with Gaussian mixture models. For the interviews, we had two goals in mind: First, assessing the practical usefulness of the three design concepts for the experts, and second, evaluating how the design concepts contribute to the understanding of Gaussian mixture models with respect to the four domain independent user needs N1–N4.

### 6.1. Setup and analysis of the interviews

We solicited feedback from five researchers and data analysts in the fields of statistical modeling, machine learning, visualization, and air traffic services. In what follows, we refer to the interviewees as experts (for Gaussian mixture models). Their experience in working with Gaussian mixture models ranges from two to ten years. All five experts work not just with Gaussian mixture models, but also with underlying data. We explained to them how data and model slices can be combined for model validation, but here we are only interested in their feedback on the visualizations of model slices, because the visualization of data slices is already well established.

The interviews were conducted using an online-conferencing tool. For later analysis, we recorded the interviews on video, except for one, where we took handwritten notes.

Within the interviews, we first asked the experts about their own work with Gaussian mixture models, before they gave us some general feedback on the design concepts that we presented to them by always using the same two example models, namely, the synthetic model with four components that is shown in the teaser, and a model that is derived from the Auto MPG data set (Quinlan, 1993). The latter model is also used in the user study that we describe in Section 7. We also discussed advantages and disadvantages of the different concepts in the experts' work context. Finally, we asked the experts to rank the design concepts.

After the interviews, we encoded and categorized the experts' answers using an open coding scheme. We also checked for potential connections to the domain-independent user needs N1–N4 from Section 2. During the interviews, these needs were never explicitly mentioned or discussed with the experts.

### 6.2. Use of and tasks on the mixture models

The experts use Gaussian mixture models for clustering, topic modeling, density estimation, dimension reduction, and visualization. Clustering is the use case most frequently mentioned, namely, by four of the five experts. All the experts expressed the need to understand models without underlying data, especially the expert from the area of machine learning, who is mostly interested in Gaussian mixture models, either as a baseline or a subroutine for other methods. Two of the experts who are working on statistical modeling and visualization, respectively, are also interested in outliers, that is, data points that are not representative for their assigned clusters, and in data points that lie between clusters. However, they understood that clustering, as well as outlier detection, are data tasks, but not model tasks.

In the analysis of the interviews, we categorized the experts' answers using an open coding scheme. From the codes, we extracted a number of recurring tasks and the experts' assessments of the design concepts with respect to the tasks. Finally, we mapped the tasks to the needs N1–N4. The tasks, extracted assessments, and the mapping of tasks to needs are summarized in Table 2.

### 6.3. Evaluating the design concepts

Three experts explicitly stated the usefulness of the simultaneous representation of the components and the whole distribution, which underlines the practical relevance of the design problem at hand. Furthermore, the tasks that we extracted from the interviews cover all four user needs N1–N4, which supports the soundness of our user needs analysis.

**Table 2**
Tasks that were reported by the experts together with a mapping to the user needs N1–N4 and the extracted assessments of how well the design concepts support the tasks. Here, + means positive sentiment, − means negative sentiment, and ○ means neither positive nor negative sentiment.

| | Experts' task | Contours | Pie Charts | Principal Axes |
|---|---|---|---|---|
| N1 | Locate mean of component | +○ | | ++ |
| | Assess shape of component (weight, variance) | +++− | − | +++ |
| | Assess probability that point belongs to component | +++− | +++++ | |
| N2 | Contrast dense/narrow versus spread components | ++○ | +− | ++ |
| | Identify overlap of components | − | | + |
| N3 | Identify global maximum | ○ | − | ○ |
| | Assess overall probability of a point | ++○ | ○− | + |
| N4 | Identify number of components | +○− | − | + |
| | Identify regions of highest influence | +− | +− | + |

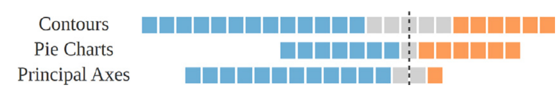


**Fig. 12.** Visual summarization of the experts' assessments of the design concepts. The horizontal bars encode the number of positive (blue), neutral (gray), and negative (orange) assessments, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

All three design concepts are considered useful by the experts. However, their usefulness depends strongly on the task, as can be seen in Table 2 and Fig. 12.

Overall, that is, aggregated over all tasks, the principal axes and the contours concepts were liked best. The principal axes concept is the preferred concept for seven out of the nine tasks. It's main strength, as seen by three experts, is supporting the assessment and comparison of the components' features, while its main weakness in comparison to the contours and especially to the pie chart concepts is support for assessing the probability that a point belongs to a component. The contours concept is preferred for assessing the overall probability. The pie charts concept is unanimously considered the best suited for assessing the probability that a point belongs to a component, while it received some critical feedback for tasks related to the shape of a distribution.

The experts' direct ranking of the design concepts is summarized in Table 3. The stated rankings correlate well with the preferences for design concepts on tasks that we independently derived from the coding scheme in Table 2. Scoring the design concepts by counting the number of positive sentiments and subtracting the number of negative sentiments ranks the principal axes concept first (Score 11), the contours concept second (Score 7), and the pie charts concept third (Score 1), whereas the contours concept is slightly ahead of the principal axes concept in the expert's median ranking.

Three experts named a single best design concept, whereas two experts ranked several concepts equally. Also, when asked what concepts they would like to work with, four out of five experts named both the contours and the pie charts concept. Therefore, we conclude that there is indeed not a single best concept, but all concepts jointly contribute to the understanding of Gaussian mixture models. Fig. 11 shows a simple yet effective approach to how different concepts can be combined for the visualization of higher-dimensional Gaussian mixture models.

**Table 3**
Preference ranking of the domain experts for the design concepts. If an expert ranked multiple concepts equally, then we reported the mean of the collapsed ranks. For instance, Expert 3 ranked all three concepts equally, and thus we assigned the rank $(1+2+3)/3 = 2$ to all three concepts.

| | Median rank | Expert 1 | Expert 2 | Expert 3 | Expert 4 | Expert 5 |
|---|---|---|---|---|---|---|
| Contours | 1.5 | 2 | 1.5 | 2 | 1 | 1 |
| Pie charts | 3 | 3 | 1.5 | 2 | 3 | 3 |
| Principal axes | 2 | 1 | 3 | 2 | 2 | 2 |

### 6.4. General feedback

All experts stated that they want an interactive tool. Specific suggestions for such a tool include a facility for choosing the number and values of isolevels in contour plots, especially for representing the whole distribution. This could improve the assessment of the shape of the whole distribution and how it is influenced by the components. One expert suggested a flexible tool that allows to choose the design for the whole model and its components independently. Furthermore, one expert proposed to combine the positive aspects of the pie chart concept with one of the other concepts by showing a pie chart at an interactively chosen point.

## 7. Insight-based user study

We complemented the expert interviews with an insight-based user study with non-expert users. Insight-based evaluation methods measure the understanding of data by counting the number of correct insights that participants gained from a certain visualization. It avoids forcing participants to perform unnatural benchmark tasks (North et al., 2011), that is, tasks that are well-defined and measurable but miss the point of the problem at hand. Here, using these insight-based methods allows us to measure how much a design concept encourages insights about a certain insight category (in our case N1–N4) as well as to find additional important categories, that might have not been considered before (North et al., 2011). Hence, in our between-subjects empirical experiment, each participant was shown a single visualization of the Gaussian mixture model and asked to report their insights about it. We analyzed

(A) the effect of a design concept on people's ability to gain correct insights, and
(B) the type of insight a concept encourages in the context of understanding Gaussian mixture models.

*Experimental setup*

We recruited 79 students and researchers in science, engineering, and mathematics for this experiment. This group of participants was chosen, because the task requires at least a basic understanding of probability density functions. We removed the answers of ten participants from the analysis, because they did answer in another language or did not give any answers about the model shown. This leaves us with 69 participants (48 male, 20 female, 1 diverse), aged 21 to 38 years (mean: 25.4 years). 49.5% of them reported to be at least somehow familiar with probability density functions. All but three participants reported normal or corrected to normal vision. Participation was voluntary without reward, however, participants received a summary of the study's results.

The overall study design, originally intended as a pure online study, was improved by a pilot study with five of the authors' colleagues who did not participate in the subsequent study. Besides improvements in wording and format, we noted that most of the researchers had a tough time remembering the concept of probability density functions. Therefore, we decided to conduct a lab study so that we had the opportunity to briefly teach the participants about probability density functions in general and Gaussian mixture models in particular. Participants also had the chance to ask questions. After the introduction, participants individually performed the following tasks on a personal computer: First, they had to complete an informed consent form and calibrate their monitor to ensure uniform stimulus sizes. Then, the participants received an explanation about the design concept that was randomly assigned to them. Next, participants were confronted with the corresponding stimulus (see Fig. 13 for the stimuli we used in this experiment) and their task: "*Please describe the mixture distribution that is visualized here! We are interested in everything you notice (all qualities, patterns, structures, and anomalies). This is about your personal opinion and not about right or wrong answers.*"

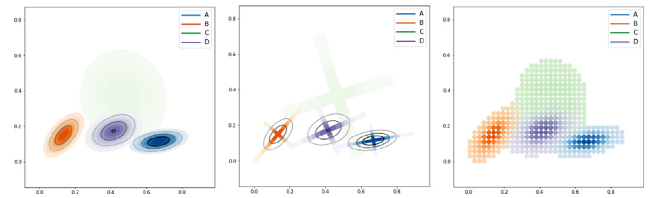

**Fig. 13.** The three stimuli used in our experiment. They visualize a Gaussian mixture model derived from the AUTO MPG data set (Quinlan, 1993). Each participant received one of the stimuli, according to the assigned group.

The time for completing the task was not limited as we wanted the participants to invest as much effort as they considered appropriate. The study ended with a post-experiment questionnaire asking the respondents for their age, gender, vision, usual length of self-written texts, familiarity with probability density functions, point of contact with these, and their familiarity with the presented concepts.

*Analysis*

In order to evaluate the effects of the three design concepts concerning their stimulation of correct insights and insights about the user needs N1–N4, we encoded the participants' insights, computed 95% confidence intervals, and interpreted these. We started by dividing the participants' answers into insights, that is, single observations about the model. Next, we encoded the insights according to two code books. The first code book categorized insights according to their correctness (true, false, miscellany). The second codebook had the categories: insight about components (Need N1), insight about the relation among components (Need N2), insight about the whole distribution (Need N3), insight about the relation of the components to the whole (Need N4), insight about the visualization, and miscellany. The miscellany category contains all insights that do not fit into any of the other categories. In the first coding round, three of us individually assigned categories to the insights, and we discussed and resolved differing assignments in the second round. Then we computed 95% confidence intervals using a bootstrapping approach. We drew 10,000 bootstrap samples uniformly at random from the available answers, computed their point estimates, and identified the 95% confidence intervals by the percentile method (Efron and Tibshirani, 1986). This way we computed the point estimates and their confidence intervals for (1) the average proportions of the correctness categories, (2) the average number of correct insights for the user needs categories, and (3) the corresponding differences between the design concepts.

If one of the difference confidence intervals does not contain the value zero, then the difference between the two corresponding concepts can be considered significant at a significance level of $\alpha = 0.05$. For instance, consider the confidence interval for the difference $A - B$. If its lower bound is greater than 0 then the difference is significant in favor of $A$.

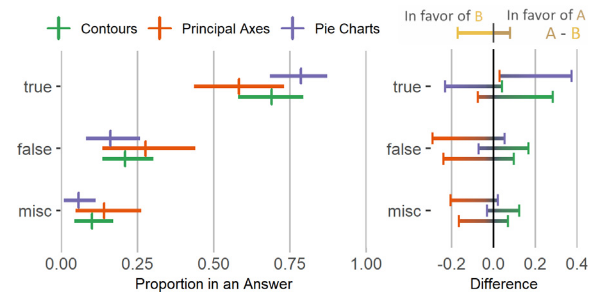


**Fig. 14.** Average proportion of the correctness categories in the overall response and the differences between the design concepts together with 95% confidence intervals based on 10 000 bootstrap samples.

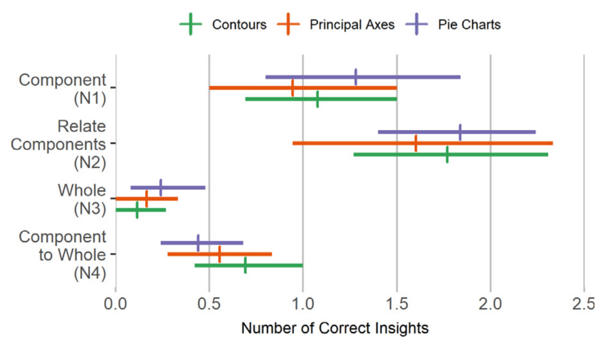


**Fig. 15.** Average number of correct insights per participant for each user needs category together with 95% confidence intervals based on 10 000 bootstrap samples.

*Results and discussion*

**Ability to gain correct insights.** The minimum, average, and maximum number of insights in the participants' answers are 1, 5.01, and 10, respectively, with 70% of them being correct, 21% being wrong, and 9% are comments that were not about the underlying model. The average number of insights varies only marginally between the three design concepts (Contours: 5.15, Principal Axes: 5.22, Pie Charts: 4.72) without any significant difference.

Fig. 14 clearly shows that each design concept successfully facilitates the understanding of a Gaussian mixture model, since all the point estimates for the proportion of correct insights in an answer are between 58% and 78%, which is significantly higher than the proportion of non-correct answers. The proportion of correct insights in the answers was highest with the pie charts and lowest with the principal axes design concept. The difference between the principal axes and the pie chart concepts is even significant, since the corresponding confidence interval does not contain the value 0.

The tendency of the principal axes concept to stimulate fewer correct answers and at the same time more wrong answers as well as more answers about the visualization indicates some confusion about this design concept. Remember, that the principal axes concept encodes the parameters of the Gaussian components, instead of their density functions. Hence, it probably requires a deeper understanding of the parameters represented by the principal axes in order to stimulate proper insights about the model. It seems the principal axes concept is not an ideal choice for non-expert users.

**Type of insights.** With each design concept there is a similar number of correct insights in each user needs category (see Fig. 15). Hence, each user need seems, in contrast to our expectations, similarly well represented by the three design concepts.

The design concepts stimulate an average number of 3.65, 3.51, and 3.64 insights about the parts of a mixture model (N1 + N2) and an average number of 0.96, 0.78, and 0.76 insights about the whole distribution (N3 + N4) with highly overlapping confidence intervals. This shows that all design concepts represent the whole and the parts equally well. The larger number of insights about the individual components and their relations can mostly be explained by the number of probability density functions. It seems logical that four probability density functions (the components) stimulate more insights than a single function (the whole). However, comments like "*It is also not directly obvious to me why 'green' does not have circles around the color origin*" and "*I forgot what the black rings meant in detail, …*" indicate that the concept of the whole distribution or its visual representation remained unclear to some of the participants.

Typical insights from the participants' answers related to components (Need N1) are about the components' mean ("*Component A has the highest density approximately at* $x = 6.5$ *and* $y = 0.15$"), the components' overall density ("*The density in C is low everywhere*"), and the components' variance ( "*A-blue is more elongated in the x dimension.*").

Typical insights about the relations among components (Need N2) compare the components' means ("*The mean value of* $p_B(x)$ *is not as high* $p_D(x)$ *and* $p_A(x)$*.*"), their variances ("*The variance of the single distributions seems to be similar for A, B, D, while it is much larger for C.*"), and identified overlapping regions ("*Purple and green are the most mixed.*").

There are only a few insights about the whole distribution (Need N3). These insights concern the number of modes ("*The diagram has three 'strong' peaks.*"), their position ("*The peaks are distributed approximately equally along the x-axis.*"), and their variance ("*The overall variance in the x-direction is much greater than that in the y-direction.*").

Insights from the participants' answers that are about the relation of the components to the whole (Need N4) are mostly about the number of components ("*Mixture of four probability densities*") and about the contributions of individual components to the whole density ("*B, D and A probably have a larger share in the whole distribution*").

Finally, typical insights about the visualization instead of the model describe visual elements ("*The four probabilities are represented by differently colored circles.*"), complain about the readability of the visualization and its colors ( "*It is hard to read the diagram.*" ), and wondered about the meaning of specific visual elements ("*I can no longer say what the orientation of the single distributions should represent.*"). As mentioned before we consider insights/comments like this as an indication that the visualization is difficult to process and distracts from understanding the visualized model.

In summary, the insight-based study found evidence that (1) all design concepts successfully promote the understanding of Gaussian mixture models and represent the whole and the parts equally well, and that (2) the principal axes concept causes some confusion for non-experts, while the pie charts concept outperforms it.

## 8. Limitations and future work

Here, we have focused on the *visual* design space for Gaussian mixture models and so far omitted *interaction* in order to keep the concerns of visual design and interaction design separate. However, we consider the transformation of the static design concepts into an interactive tool as the main direction of the future work. Based on the insights from the expert interviews in Section 6 and the user study in Section 7 that the concepts have complementary strengths and weaknesses, that is, they are far from equal when it comes to specific tasks, we believe that a tool that combines multiple design concepts and adds interactivity such as zooming and panning as well as the interactive choice of isolevels/values and pie chart resolution will provide a real benefit to practitioners. Any practically useful tool must also support the simultaneous visualization of data that underlies the mixture model (Lawonn et al., 2022). Fortunately, this can be achieved easily, because the slicing approach applies simultaneously to both multivariate models (marginalization) and multidimensional data (coordinate projections). Therefore, the problem basically reduces to finding a good way to overlay attributed scatter plots over two-dimensional marginal plots.

For Gaussian mixture models that exhibit regions with many, significantly contributing (overlapping) components, the three design concepts may result in visual clutter. Interaction is an obvious option to address this problem. However, regions with a high number of contributing components seem to be rare in real-world applications (Kawabata, 2008; Li et al., 2016; Shin et al., 2009).

Finally, we want to point out an interesting problem for high-dimensional models with many components, namely, how to assign a limited set of colors to the components such that the components can be distinguished well in any two-dimensional marginal plot. For a single, two-dimensional marginal plot, the problem can be addressed, for instance, by applying the four-color theorem to a Voronoi tessellation of the component's mean vectors, but the high-dimensional case still poses a challenge.

## 9. Conclusions

The correspondence between the data slicing operations of projection and selection and the probabilistic model slicing operations of marginalization and conditioning makes it possible to adopt the slicing-based visual data exploration approach also for probabilistic models. This, however, requires us to come up with a visualization design for model slices. Here, we have explored the design space for a particular type of model slices, namely, mixture model slices, or more specifically Gaussian mixture models. Mixture model slices are particularly interesting for model validation, because they showcase interactions between categorical and continuous variables. The design space for visualizations of Gaussian mixture models is large and some guiding principles are needed for choosing a good design. Here, we derived such guiding principles from four common, application-domain independent, practically relevant needs, namely, (N1) understanding the individual components of a Gaussian mixture model, (N2) relating the components to each other, (N3) understanding the whole mixture distribution, and (N4) relating the components to the whole distribution. To address these needs, we have developed and discussed three design concepts for the visualization of Gaussian mixture models. Our evaluation of the three design concepts by an insight-based user study found that the three fairly different design concepts support the identified needs (N1–N4) and the overall understanding of Gaussian mixture models similarly well. This finding is backed by feedback from expert users. However, the experts also provided the additional insight that the concepts have complementary strengths and weaknesses, that is, they are not equal, when it comes to specific tasks. Only tasks related to N1 are supported well by all three design concepts. However, the contours design concept supports more tasks related to N1 equally well than the other two concepts. Tasks related to N2–N4 are not well supported by the pie chart design concept. The pie charts concept, however, is best suited for the task of assessing the probability that a point belongs to a component, which relates to N1. It also triggered more correct and fewer false insights than the other design concepts in our user study with non-expert users. The principal components design, which fared well in the experts' evaluation, triggered significantly fewer correct and significantly more false insights. Therefore, we conclude that there is not a single best design and recommend working with all three concepts simultaneously.

## CRediT authorship contribution statement

**Joachim Giesen:** Conceptualization, Formal Analysis, Writing – review & editing. **Philipp Lucas:** Conceptualization, Programming, Evaluation, Writing. **Linda Pfeiffer:** Conceptualization, Evaluation, Writing. **Laines Schmalwasser:** Conceptualization, Programming, Writing. **Kai Lawonn:** Writing – review & editing, Supervision.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Ethical Approval

This study does not contain any studies with human or animal subjects performed by any of the authors.

## Acknowledgments

This work was supported by the Carl Zeiss Foundation, within the project Interactive Inference.

## References

Allili, M.S., Bouguila, N., Ziou, D., 2007. Finite generalized Gaussian mixture modeling and applications to image and video foreground segmentation. In: Canadian Conference on Computer and Robot Vision. CRV, IEEE Computer Society, pp. 183–190.
Aubert, A.H., Thrun, M.C., Breuer, L., Ultsch, A., 2016. Knowledge discovery from high-frequency stream nitrate concentrations: hydrology and biology contributions. Sci. Rep. 6.
Bernaille, L., Teixeira, R., Salamatian, K., 2006. Early application identification. In: Diot, C., Ammar, M.H. (Eds.), Proceedings of the Conference on Emerging Network Experiment and Technology. CoNEXT, ACM, p. 6.
Bishop, C.M., 2007. Pattern recognition and machine learning, 5th edition. In: Information science and statistics, Springer.
Brewer, C.A., 2013. ColorBrewer 2.0: color advice for cartography. https://colorbrewer2.org.
Brewer, C., Campbell, A.J., 1998. Beyond graduated circles: Varied point symbols for representing quantitative data on maps. Cartograp. Perspect. 29, 6–25.
Chen, M., Walton, S.J., Berger, K., Thiyagalingam, J., Duffy, B., Fang, H., Holloway, C., Trefethen, A.E., 2014. Visual multiplexing. Comput. Graph. Forum 33 (3), 241–250.
Dempster, A., 1973. Aspects of the multinomial logit model. In: Third International Symposium on Multivariate Analysis. Academic Press, pp. 129–142.
Efron, B., Tibshirani, R., 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statist. Sci. 1 (1), 54–75.
Friendly, M., Denis, D., 2005. The early origins and development of the scatterplot. J. History Behav. Sci. 41 (2), 103–130.
Harrower, M., Brewer, C.A., 2003. ColorBrewer.org: An online tool for selecting colour schemes for maps. Cartograph. J. 40 (1), 27–37.
Hartigan, J., 1975. Printer graphics for clustering. J. Stat. Comput. Simul. 4 (3), 187–213.
Inselberg, A., Dimsdale, B., 1990. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In: IEEE Visual.. IEEE, pp. 361–378.
Kawabata, T., 2008. Multiple subunit fitting into a low-resolution density map of a macromolecular complex using a Gaussian mixture model. Biophys. J. 95, 4643–4658.
Klaus, J., Blacher, M., Goral, A., Lucas, P., Giesen, J., 2023. A visual analytics workflow for probabilistic modeling. Visual Inf. 7 (2), 72–84.
Lawonn, K., Meuschke, M., Eulzer, P., Mitterreiter, M., Giesen, J., Günther, T., 2022. GRay: Ray casting for visualization and interactive data exploration of Gaussian mixture models. In: IEEE Transactions on Visualization and Computer Graphics. in print.

Lee, K.J., Guillemot, L., Yue, Y.L., Kramer, M., Champion, D.J., 2012. Application of the Gaussian mixture model in pulsar astronomy - pulsar classification and candidates ranking for the Fermi 2FGL catalogue. Mon. Not. R. Astron. Soc. 424 (4), 2832–2840.

Li, L., Hansman, R., Palacios, R., Welsch, R., 2016. Anomaly detection via a Gaussian mixture model for flight operation and safety monitoring. Transp. Res. C 64, 45–57.

Liu, S., Levine, J.A., Bremer, P., Pascucci, V., 2012. Gaussian mixture model based volume visualization. In: Barga, R.S., Pfister, H., Rogers, D.H. (Eds.), Symposium on Large Data Analysis and Visualization. LDAV, IEEE Computer Society, pp. 73–77.

Mathur, N., Asirvadam, V., Dass, S., 2018. Spatial-temporal visualization of dengue incidences using Gaussian kernel. In: International Conference on Intelligent and Advanced System. ICIAS.

Mayorga, A., Gleicher, M., 2013. Splatterplots: Overcoming overdraw in scatter plots. IEEE Trans. Visual. Comput. Graph. 19 (9), 1526–1538.

Munzner, T., 2015. Visualization analysis and design. In: A.K. Peters visualization series, CRC Press, Taylor & Francis Group, URL http://www.cs.ubc.ca/%7Etmm/vadbook/.

North, C., Saraiya, P., Duca, K., 2011. A comparison of benchmark task and insight evaluation methods for information visualization. Inf. Visual. 10 (3), 162–181.

Olkin, I., Tate, R., 1961. Multivariate correlation models with mixed discrete and continuous variables. Ann. Math. Stat. 32 (4), 448–465.

Porter, T.K., Duff, T., 1984. Compositing digital images. In: Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH, pp. 253–259.

Quinlan, J.R., 1993. Combining instance-based and model-based learning. In: International Conference on Machine Learning. ICML, pp. 236–243.

Sarikaya, A., Gleicher, M., 2018. Scatterplots: Tasks, data, and designs. IEEE Trans. Vis. Comput. Graphics 24 (1), 402–412.

Shin, M.-S., Sekora, M., Byun, Y.-I., 2009. Detecting variability in massive astronomical time-series data. I: application of an infinite Gaussian mixture model. Mon. Not. R. Astron. Soc. 400 (4), 1897–1910.

Stolte, C., Tang, D., Hanrahan, P., 2002a. Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. IEEE Trans. Vis. Comput. Graphics 8 (1), 52–65.

Stolte, C., Tang, D., Hanrahan, P., 2002b. Query, analysis, and visualization of hierarchically structured data using polaris. In: International Conference on Knowledge Discovery and Data Mining. SIGKDD, pp. 112–122.

Talbot, J., Setlur, V., Anand, A., 2014. Four Experiments on the Perception of Bar Charts. IEEE Trans. Viual. Comput. Graphics 20 (12), 2152–2160.

The National Climate Database, 2021. Weather data for auckland, New Zealand. https://medium.com/plotly/what-is-a-splom-chart-make-scatterplot-matrices-in-python-8dc4998921c3. (Accessed on 27 March 2021 via).

Torsney-Weir, T., Sedlmair, M., Möller, T., 2017. Sliceplorer: 1D slices for multi-dimensional continuous functions. Comput. Graph. Forum 36 (3), 167–177.

Wang, Y., Chen, W., Shan, G., Dong, T., Chi, X., 2010. Volume exploration using ellipsoidal Gaussian transfer functions. In: Pacific Visualization Symposium (PacificVis). IEEE Computer Society, pp. 25–32.

Wilkinson, L., Anand, A., Grossman, R.L., 2005. Graph-theoretic scagnostics. In: IEEE Symposium on Information Visualization. InfoVis, pp. 157–164.

Yang, L., Cheung, N., Li, J., Fang, J., 2019. Deep clustering by Gaussian mixture variational autoencoders with graph embedding. In: International Conference on Computer Vision. ICCV, pp. 6439–6448.