Eurographics Conference on Visualization (EuroVis) 2025
W. Aigner, N. Andrienko, and B. Wang
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (*2025*), *Number 3*

PrismBreak: Exploration of Multi-Dimensional Mixture Models

B. Zahoransky[1], T. Günther[2] and K. Lawonn[1]

[1]Friedrich Schiller University Jena, Germany
[2]Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

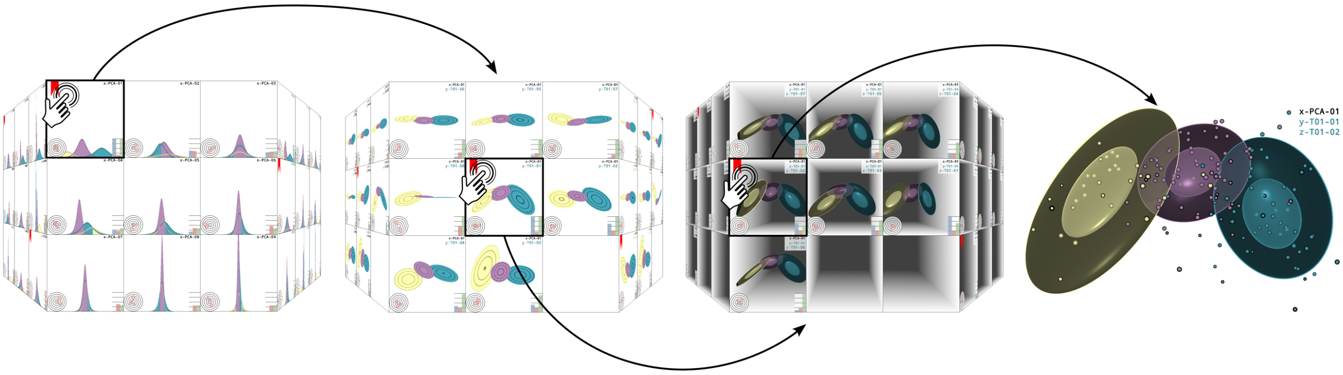


Figure 1: *We present an interactive multi-faceted prism view, which supports data scientists in the subspace exploration of multi-variate distributions. The user composes a low-dimensional space by incrementally choosing their own basis vectors. For each choice, multiple options are visualized qualitatively and are supported with quantitative information regarding variance, sparsity, and visibility. From left to right: (1) visualization of data distributions when selecting the first axis, (2) display of two-dimensional distributions, (3) three-dimensional visualization for choosing the third dimension, and lastly (4) a detailed view in which individual data points are shown.*

Abstract

In data science, visual data exploration becomes increasingly more challenging due to the continued rapid increase of data dimensionality and data sizes. To manage complexity, two orthogonal approaches are commonly used in practice: First, data is frequently clustered in high-dimensional space by fitting mixture models composed of normal distributions or Student t-distributions. Second, dimensionality reduction is employed to embed high-dimensional point clouds in a two- or three-dimensional space. Those algorithms determine the spatial arrangement in low-dimensional space without further user interaction. This leaves little room for a guided exploration and data analysis. In this paper, we propose a novel visualization system for the effective exploration and construction of potential subspaces onto which mixture models can be projected. The subspaces are spanned linearly via basis vectors, for which a vast number of basis vector combinations is theoretically imaginable. Our system guides the user step-by-step through the selection process by letting users choose one basis vector at a time. To guide the process, multiple choices are pre-visualized at once on a multi-faceted prism. In addition to the qualitative visualization of the distributions, multiple quantitative metrics are calculated by which subspaces can be compared and reordered, including variance, sparsity, and visibility. Further, a bookmarking tool lets users record and compare different basis vector combinations. The usability of the system is evaluated by data scientists and is tested on several high-dimensional data sets.

1. Introduction

The effective analysis of high-dimensional abstract data is a long-standing challenge across many application domains [Kaw08, SSB09, LHPW16, DDH*25]. In general, high-dimensional abstract data can be considered a high-dimensional point cloud, where the particular distribution of the high-dimensional points is of great interest in inferring the semantics and structures in the data. Conceptually, such data can be represented in tabular form, and hence the complexity of such a data set can be measured by two numbers: the total number of points/items (number of rows), and the number of dimensions/attributes in which the points live (number of columns) [Mun14a]. To reduce the number of data points, experts aggregate points into clusters. The point distribution inside a cluster can then be summarized with few parameters by fitting one or multiple suitable distributions to the points, such as normal distributions or Student t-distributions. A model-based analysis then focuses on the arrangement and shape of clusters, while a data-based analysis concentrates on the properties of individual data points and their cluster memberships.

We design a visualization system that supports users in the exploration of different subspaces onto which high-dimensional distributions can be projected. These subspaces are spanned by up to three orthonormal basis vectors. Previous work [LME*23] proposed a first ray tracing system that visualized high-dimensional normal distributions for given subspaces. In their work, the subspace was formulated as a linear combination of basis vectors, which were weighted using sliders. We expand their work by designing a system that guides the user through the selection process of different basis vectors, which are added incrementally. Each choice of basis vector is supported by displaying multiple possible subspaces on a multi-faceted prism. Thereby, a number of model-based and data-based analysis tasks are supported. For example, in addition to qualitative visualizations of the resulting distributions, we derive quantitative metrics, including variance, sparsity, and visibility, by which the subspaces can be compared and arranged. Further, bookmarks enable customized comparisons. The ray-tracing-based rendering of the distributions is GPU-accelerated. Since the choice of distribution is data- and application-dependent, our system generalizes to different types of distribution functions. Thus, in addition to normal distributions [LME*23], we also support Student t-distributions, which were requested by our users. Our tool is designed and evaluated with data science practitioners on several high-dimensional data sets. In summary, we contribute:

- a multi-faceted prism view that enables users to compose and compare different subspaces for dimensionality reduction,
- a set of metrics by which the subspace projections are compared, namely variance, sparsity, and visibility,
- closed-form expressions for rendering Student t-distributions,
- a user study with domain scientists,
- a novel system that is available as an open-source project [ZGL].

2. Related Work

In the following, we summarize work on multi-dimensional visualization, linear coordinate representations, exploration and interaction, and dimensionality reduction. For comprehensive entries into visualizations tailored for high-dimensional data, we refer to Liu et al. [LMW*17]. They delineated the visualization of multidimensional data into three key phases. In the *data transformation* phase, information gets reduced via topological data analysis [WSPVJ11], projections [PEP*11], and clustering [LT15]. The second phase is the *visual mapping*, in which multiple attributes are mapped to specific spatial axis arrangements [LT13, CvW11], they may also be mapped to glyphs [War08], or they are explored by animations [EDF08] and hierarchical views [OHJ*11]. The *view transformation* is the final phase, which aims to reduce visual clutter [AdOL04] or supports the viewer with additional shading effects [SW09, MG13].

Seo and Shneiderman [SS04] introduced a rank-by-feature framework to aid users in identifying pertinent dimensions for further exploration through scatterplots. Sips et al. [SNLH09] developed the notion of class consistency to enhance the exploration process. Furthermore, Tatu et al. [TAE*09] demonstrated automatic analysis for uncovering structures within high-dimensional data, thus providing users with recommendations for additional exploration. Complementing these efforts, Bertini et al. [BTK11] offered a comprehensive review of quality metrics to facilitate the navigation and exploration of complex datasets.

Next, we provide an overview of dimensionality reduction methods. Among the most popular choices for non-linear dimensionality reduction is the t-distributed Stochastic Neighborhood Embedding (t-SNE) [VdMH08]. To preserve neighborhoods in the high-dimensional space, normal distributions are fitted to the data in the high-dimensional space, which are then mapped to t-distributions in the lower-dimensional space which utilizes their heavier tails to fit in the larger region of the high-dimensional space. The similarity between the distributions is measured by the Kullback-Leibler (KL) divergence, and the variance of the distributions is chosen that each distribution has a user-chosen perplexity, analogous to a neighborhood size parameter. The algorithm produces clusters in the resulting scatter plot depending on the chosen perplexity [WVJ16]. With efficient GPU implementations, t-SNE has been optimized for high performance [CRHC18, PTM*20], and has now found numerous applications, such as in cell biology [GHS*19]. The Uniform Manifold Approximation and Projection (UMAP) [MHM18] is another popular approach that has found widespread application [BMH*19]. Like t-SNE, UMAP projects high-dimensional data to lower dimensions by estimating the underlying manifold structure while aiming to preserve both local and global structures. The Local Affine Multidimensional Projection (LAMP) [JCC*11] utilizes a subset of samples, so-called control points, and their positions in the visual space. These control points, which can be interactively rearranged, are used to construct a family of orthogonal affine mappings, assigning one mapping to each point. Principal Component Analysis (PCA) [Hot33] is a statistical dimensionality reduction method that determines a set of orthogonal basis vectors, which can be sorted in descending order by the amount of data variance they capture. The basis vectors arise as eigenvectors of the covariance matrix of the centered data points. Eigenvalues indicate variance along eigenvectors, helping assess information loss when dropped. Discarding eigenvectors with low corresponding eigenvalue enables a lower-order approximation while preserving maximum information.

3. Visualization Design

The visualization system that we propose in this paper is developed in close collaboration with domain scientists, namely visualization experts, machine learning researchers, and statisticians. In their daily work, they frequently explore and analyze high-dimensional data statistically; that is, they fit distributions and investigate their properties. Before we proceed formulating concrete analysis tasks, we need to answer two fundamental design questions.

3.1. Design Questions

Which Distributions Should Be Considered? To cluster data points, the experts we worked with use mixture models based on two distributions.

First, the k-dimensional normal distribution is parametrized by a mean vector $\mu \in \mathbb{R}^k$ and a positive-definite covariance matrix $\Sigma \in \mathbb{R}^{k\times k}$. Its probability density function is called Gaussian and is defined as

$$g(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k \det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\mu)\right).$$

Gaussians are commonly used to approximate data due to their favorable mathematical properties, such as symmetry, differentiability, and controllable compact support.

Second, the Student t-distribution is a parametric distribution that, for specific shape parameter choices $\nu \in \mathbb{R}_{>0}$, includes the standard normal distribution ($\nu \to \infty$). With the mean $\mu \in \mathbb{R}^k$ and the positive-definite scale matrix $\Sigma \in \mathbb{R}^{k\times k}$, its density is defined as

$$t(\mathbf{x}) = \frac{\Gamma\left(\frac{\nu+k}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{(\nu\pi)^k \det(\Sigma)}} \left(1+\frac{1}{\nu}(\mathbf{x}-\mu)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\mu)\right)^{-\frac{\nu+k}{2}}.$$

$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\mathrm{d}t$ represents the gamma function. In this paper, we refer to $t(\mathbf{x})$ as t-density.

The probability density function of a multi-variate mixture model is a linear combination of $n \in \mathbb{N}$ density functions, which are weighted by corresponding weights $\phi_i \in \mathbb{R}$:

$$p(\mathbf{x}) = \sum_{i=1}^{n} \phi_i \cdot f_i(\mathbf{x}). \tag{1}$$

The component density functions $f_i(\mathbf{x})$ are all treated as a Gaussian $g(\mathbf{x})$ or as a t-density $t(\mathbf{x})$ and are defined by their own parameters μ_i, Σ_i and optionally ν_i. The visual analysis of mixture models is complex since new modes are created from superpositions [LME*23, GLP*24]. For $k > 3$, the multi-variate distributions (and, as a consequence, their mixture models) are too high-dimensional to visualize directly.

Which Projection Method Is Suitable? The critical challenge that we address in this paper is the much-needed scalability of mixture model analyses concerning many dimensions. The objects studied by our domain scientists are distributions, which pose a hard constraint on any projection method: The continuity of the density function must be retained after projection; in other words, a Gaussian in high-dimensional space should look like a Gaussian after projection, as shown in Fig. 2. Non-linear dimensionality reduction methods, such as t-SNE and UMAP, are not applicable since they are prone to break connected regions into multiple pieces [BGG20]. LAMP preserves the global structure better. However, t-SNE, UMAP and LAMP are designed to reduce the dimensionality of point clouds. They are not well suited to render density functions, since it would be computationally expensive to determine the projection at each pixel based on the surrounding points. Instead, we need a unique global projection. Thus, we decided to use linear projections, not only because they retain continuity of functions and are scalable in terms of compute time, but also because there is a higher chance for interpretability since every point is formed from a linear combination of the original attributes. Since the term of *interpretability* has proven to be ambiguous [Par22, ZBLB24], we want to clarify that we aim to provide *interpretability* for the subsequent data exploration task, i.e., our approach is designed to help users understand how certain attributes contribute to the projection. However, finding a suitable linear basis and interpreting it is challenging.

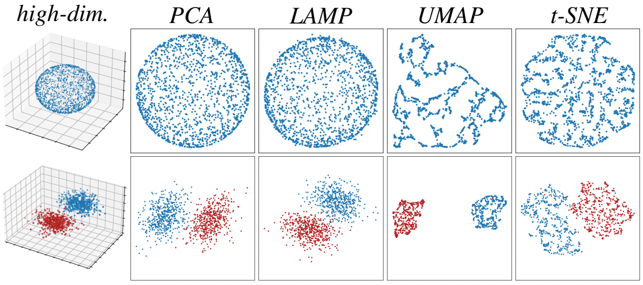


Figure 2: *Top: Uniformly distributed points on a manifold in $\mathbb{R}^3$, bottom: points follow two normal distributions in $\mathbb{R}^3$. Unlike PCA and LAMP, UMAP and t-SNE tend to fragment continuous regions (top) and do not retain the point distributions' shape (bottom).*

3.2. Problem Statement

Formal Setup. In the following, we refer to a high-dimensional data point as $\mathbf{x} \in \mathbb{R}^k$, where k is the dimensionality of the high-dimensional space. Since human visual perception is restricted to three spatial dimensions, we aim to find a linear projection that maps a point $\mathbf{x}$ to a three-dimensional coordinate $\mathbf{p} = (p_1, p_2, p_3)^{\mathrm{T}} \in \mathbb{R}^3$. The three-dimensional space is spanned by three orthonormal basis vectors $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ in $\mathbb{R}^k$. Similar to Lawonn et al. [LME*23], we introduce a linear transformation matrix by stacking the basis vectors column-wise into a matrix $\mathbf{B} \in \mathbb{R}^{k\times 3}$:

$$\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3). \tag{2}$$

The point $\mathbf{p}$ is lifted by matrix multiplication into the high-dimensional space, i.e., $\mathbf{x} = \mathbf{B}\mathbf{p}$. We refer to this as *projecting up*. Conversely, a high-dimensional point can be projected onto the three orthonormal basis vectors, which is likewise written as matrix multiplication $\mathbf{p} = \mathbf{B}^{\mathrm{T}}\mathbf{x}$. We refer to this as *projecting down*. Projecting up and back down leads to the same place where we started: $\mathbf{p} = \mathbf{B}^{\mathrm{T}}\mathbf{B}\mathbf{p}$. The reverse is not necessarily the case: $\mathbf{x} \neq \mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{x}$. Likewise, entire density functions can be projected up and down, which we explain at the example of Student t-distributions. Given a three-dimensional point $\mathbf{p}$, we can *project* this point *up* into the high-dimensional space and sample the density function $f(\mathbf{x})$:

$$f(\mathbf{B}\mathbf{p}) = \frac{\Gamma\left(\frac{\nu+k}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{(\nu\pi)^k \det(\Sigma)}} \left(1+\frac{1}{\nu}(\mathbf{B}\mathbf{p}-\mu)^{\mathrm{T}}\Sigma^{-1}(\mathbf{B}\mathbf{p}-\mu)\right)^{-\frac{\nu+k}{2}}. \tag{3}$$

This approach is computationally more costly, but has the advantage that separate structures in high-dimensional space remain separable. With this approach we only see a small subspace of a density

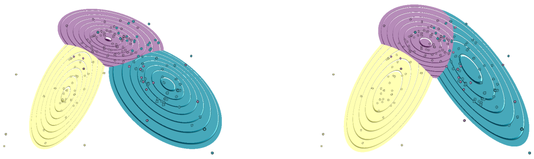


Figure 3: *Lawonn et al. [LME*23] projected points down and sampled distributions in high-dimensional space (left). We also allow projecting distributions down to low-dimensional space (right).*

function. Alternatively, we may *project* the mean $\widehat{\mu} = \mathbf{B}^{\mathrm{T}}\mu$ and the covariance matrix $\widehat{\Sigma} = \mathbf{B}^{\mathrm{T}}\Sigma\mathbf{B}$ *down*. This gives rise to a density function $\widehat{f}(\mathbf{p})$ in three dimensions:

$$\widehat{f}(\mathbf{p}) = \frac{\Gamma\left(\frac{\nu+k}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{(\nu\pi)^k \det(\mathbf{B}^{\mathrm{T}}\Sigma\mathbf{B})}} \left(1+\frac{1}{\nu}(\mathbf{p}-\mathbf{B}^{\mathrm{T}}\mu)^{\mathrm{T}}(\mathbf{B}^{\mathrm{T}}\Sigma\mathbf{B})^{-1}(\mathbf{p}-\mathbf{B}^{\mathrm{T}}\mu)\right)^{-\frac{\nu+k}{2}}. \qquad (4)$$

This formulation is computationally cheaper, but structures that are separable in high-dimensional space may be projected onto the same location in three dimensions. By default, we use this option.

Rendering of Distributions. To render Gaussians efficiently, Lawonn et al. [LME*23] derived closed-form expressions for ray integrals and maximum intensity projections using an upwards projection similar to Eq. (3). Analogously, we derive closed-form expressions for projecting t-densities up and projecting both Gaussians and t-densities down. Fig. 3 shows a t-distribution mixture model fitted to a nine-dimensional dataset and rendered with a maximum intensity projection. The results differ marginally, so we let the user decide on a mode of projection. The rendering is described in Section 4.6 and is based on the calculation of integrals and extremal points along ray segments.

3.3. Requirement Analysis

The central problem that is solved by our interactive visualization system is to guide the user in making an informed choice for the three orthonormal basis vectors $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ in Eq. (2). Since many possible basis vector combinations exist, we follow an incremental construction in which the user adds one basis vector at a time. The following tasks have been identified in collaboration with the domain scientists during contextual inquiries [WHK90].

Model-based Tasks are concerned with comparing individual distributions within a certain mixture model. The effective comparison requires a suitable choice of basis vectors.

T1 Basis Vector Pool. The system should offer a pool of reasonable basis vectors to start the exploration from. Common choices are the canonical axes (each attribute is its own axis), or principal components of the entire data set or individual distributions.

T2 Qualitative Comparison. To judge how adding a basis vector impacts the separability of distributions into clusters, it is essential to visually convey how the distributions will be arranged spatially once the next basis vector is added. This will allow for easy identification of interesting arrangements, such as outliers.

T3 Quantitative Comparison. Even within a group of basis vectors, such as the canonical basis vectors, there are potentially many possible basis vectors to choose from. To reorder and organize them, quantitative metrics are needed to judge the utility of a potential basis vector. Of particular interest are the variance captured by the basis vector, the sparseness in terms of non-zero vector entries, and the amount of occlusion caused by overlapping distributions.

T4 Vector Attribution. Basis vectors in a high-dimensional space are generally challenging to interpret semantically. It would be helpful to convey what the weight of a basis vector means in terms of the underlying attributes.

Data-based Tasks. Since a distribution is a visual representation of a group of points, it is essential to convey information about the points themselves to judge how well the distribution represents the actual data.

T5 Cluster Membership. The probability density function of each distribution indicates how likely a point at a given location is to be part of the corresponding cluster. This is referred to as fuzzy clustering. For some data points, multiple distributions might overlap, making the cluster membership less certain. It will be essential to see which cluster(s) a data point belongs to.

T6 Model Fit. The density functions inside a mixture model are an approximate stand-in for a group of data points. A model-based analysis is potentially misleading if the model does not represent the data well. Thus, it is necessary to see the individual data points relative to the density function to judge if the symmetry and shape assumptions of the distribution are valid for the data.

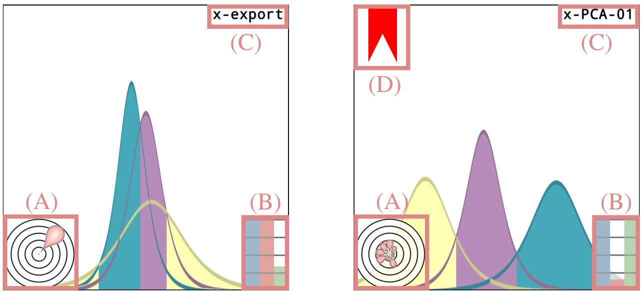
T7 Outlier Detection. When analyzing data points, detecting outliers is often critical since it allows the user to judge when and where the assumed distribution is not a good fit for the data. What constitutes an outlier depends on the analysis task. For example, a point may be considered an outlier either if it is far from the mean of its cluster or if it has an equal probability of belonging to multiple clusters.

4. Visualization System

In the following, we explain the components of our interactive visualization system, which are shown in Fig. 1. The input to our algorithm is a mixture model as in Eq. (1) with given distribution parameters. If the user provides points only, we use existing software to fit a multi-variate mixture model [AWBM18]. Further *Impementation Details* can be found in the supplemental material. Our users incrementally choose the basis vector $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$. Even for two basis vectors, presenting all possible combinations may already be infeasible. Instead, to let the user build a mental model during the selection process, we introduce the *prism view*, which applies consistent interaction metaphors throughout the incremental construction process. Each basis vector choice is supported by comparing different options with visualizations appropriately chosen for the given dimensionality of the already selected subspace.

4.1. Prism View

The starting point of the exploration is to choose the first basis vector. As mentioned in **T1**, we need a pool of potential basis vectors to start from. We chose to compose a pool from three sets of basis



Canonical Tile *Bookmarked PCA Tile*

Figure 4: *Above, we see two tiles of the first stage. The tiles show three t-distributions. There are four UI elements: (A) the attribution glyph conveys the combination of canonical basis vectors (i.e., data attributes), (B) depiction of three metrics: variance, sparsity and visibility, (C) a unique label, and (D) a custom bookmark.*

vectors, which the users frequently use:

Canonical basis vectors. Each attribute of the high-dimensional data set forms its own basis vector. This means that the number of canonical basis vectors scales linearly with the dimensionality of the data set. These basis vectors have the highest interpretability since each has a semantic meaning.

Global PCA basis vectors. A model-free basis is formed by computing a PCA across all data set points. The vectors can be ordered naturally by the variance carried by the eigenvectors. The number of information-carrying basis vectors is data-dependent, but in the worst case, it is also linear in the number of dimensions.

Per-distribution PCA basis vectors. Further, we compute a PCA based on the covariance matrix of every single distribution of the mixture model provided by the user. This way, other distributions can be viewed 'from the perspective' of one given distribution. Thus, the number of these basis vectors additionally scales linearly in the number of distributions in the mixture model.

If there are n distributions in the mixture model, the three kinds of basis vectors above naturally lead to $n+2$ groups of possible basis vectors: one group contains all canonical basis vectors, one group includes the global PCA basis vectors, and lastly, one group of PCA basis vectors for every distribution in the mixture model. This means a scalable visual encoding is needed to explore a variable number of groups and basis vectors within each group. For this reason, we designed the *prism view*. The prism view is a multifaceted prism that can be infinitely rotated horizontally. It always shows three facets at a time but can scroll through an arbitrary number of facets. Each of the groups above is assigned to one facet.

4.2. Tiles

Each facet has one tile for every basis vector that can be selected. To enable the qualitative comparison in T2, the visual encoding is adjusted to the number of basis vectors that have already been chosen. If no basis vector has been chosen, then a data histogram is shown for every potential basis vector. The histogram displays the distributions as shown in Fig. 4, allowing us to spot basis vectors for which two or more distributions are already separable. Each distribution is assigned a unique color, used from the *iWantHue* website [Mat14]. When the basis vector $\mathbf{b}_1$ has been selected, then the prism view transitions to the next stage, in which each tile displays a two-dimensional visualization of the distributions in the space spanned by $\mathbf{b}_1$ and another basis vector that can be chosen next. If the second basis vector $\mathbf{b}_2$ is chosen, we enter the third stage, in which three-dimensional distributions are shown for the space spanned by $\mathbf{b}_1$, $\mathbf{b}_2$, and the next possible basis vector. After selecting $\mathbf{b}_3$, a detailed view is shown, in which the data points can be explored. Throughout the selection process, the user can customize the color or hide distributions, enable (default) or disable scaling by weight ϕ_i (relative scaling), scale distributions, and clip distributions if their density value falls below a user-defined threshold; by default, two standard deviations of each distribution are shown. Each tile has a unique label, as shown in Fig. 4 C. Alongside the two shown options, a first stage label for the per-distribution PCA is composed of an axis, model indicator and distribution number, and basis vector number. An example is *x-G01-02* for the second principal component of the first Gaussian distribution. For t-distributions the *G* in *x-G01-02* is replaced by *T*. Every stage extends the label by a further line of the described format.

4.3. Metrics

In the following, let $\mathbf{X} \in \mathbb{R}^{N \times k}$ be the centered data matrix containing all N points in k dimensions. Further, let $\overline{\Sigma}$ be the global covariance matrix, which is symmetric and positive semi-definite:

$$\overline{\Sigma} = \frac{1}{N-1}\mathbf{X}^{\mathrm{T}}\mathbf{X},$$

with eigenvalues $\lambda_i \geq 0$, orthonormal eigenvectors $\mathbf{v}_i$, and $1 \leq i \leq k$.

Variance (◎). The first metric measures the total variance captured by a basis vector. Vectors with higher variance are preferable since there is a higher chance that the distributions will remain distinguishable. Our basis vector options included canonical basis vectors and PCA basis vectors, i.e., there are two cases:

$$s_{\text{variance}}(\mathbf{b}) = \begin{cases} \frac{\lambda_i}{\max(\{\lambda_j\}_{j=1}^{k})} & \mathbf{b} = \mathbf{v}_i \\ \frac{\sigma_i^2}{\max(\{\sigma_j^2\}_{j=1}^{k})} & \mathbf{b} = \mathbf{c}_i. \end{cases}$$

If $\mathbf{b}$ is a PCA basis vector $\mathbf{v}_i$, then the variance is expressed by the eigenvalue λ_i of the corresponding eigenvector. If $\mathbf{b}$ is a canonical basis vector $\mathbf{c}_i$, then the variance is the i-th diagonal element σ_i^2 of $\overline{\Sigma}$. In both cases, the value range is normalized to $[0,1]$ by dividing by the largest value.

Sparsity (◎). The sparsity metric relates to the interpretability of a basis vector. If a basis vector is a canonical basis vector, it corresponds to a data attribute. If the basis vector is formed as a linear combination of two canonical basis vectors, then it is a linear combination of two data attributes. A basis vector becomes more interpretable if formed from fewer canonical basis vectors. To account for how far away the values are from zero, we apply a monotonic transfer function to each basis vector component b_i in $\mathbf{b} = (b_1, \ldots, b_k)^{\mathrm{T}}$ and sum up the result:

$$s_{\text{sparsity}}(\mathbf{b}) = \sum_{i=1}^{k} b_i^4.$$

The exponent assigns a higher penalty to values far from one. Note, that an exponent of two would always result in a score of one since

b is normalized. With an exponent of four, canonical basis vectors achieve a score of one; others obtain a lower score.

Visibility (◎). In Fig. 4, we display two tiles from the first stage. In each tile, the x-axis represents the domain of the mixture model, while the y-axis indicates its density. To evaluate visibility, we focus on a single row from each tile, where each pixel l corresponds to a specific x-value. At this x-value, we compute the density p_{il} for each distribution i. This process enables us to construct a matrix $\mathbf{M}(\mathbf{b}) \in \mathbb{R}^{n\times n}$ that captures the correlation between the distributions, where n is the number of distributions:

$$\mathbf{M}_{ij} = \sum_l p_{il} \cdot p_{jl}.$$

Based on $\mathbf{M}$ we introduce $s_{\text{visibility}}(\mathbf{b}) \in (0,1)$ for each tile:

$$s_{\text{visibility}}(\mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\exp(\mathbf{M}_{ii})}{\sum_{j=1}^{k}\exp(\mathbf{M}_{ij})}. \tag{5}$$

Note that each summand is the softmax function relating the visible part of a distribution to the part covered by other distributions. Thus, overlapping regions and poor visibility of distributions lower $s_{\text{visibility}}(\mathbf{b})$, while less overlap and high visibility of the distributions increase the value. This is also evident in Fig. 4. Notably, in the PCA Tile, visibility appears to reach its maximum. The $\mathbf{M}_{ii}$ values significantly surpass all $\mathbf{M}_{ij}$ for $i \neq j$, causing nearly all summands in Eq. (5) to approach one, leading the weighted sum to do the same. To calculate the visibility for stage two and three, we only modify p_{il}. In the second stage, the domain of the distribution is two-dimensional. Consequently, we evaluate visibility using all pixels of the tile. Each pixel l corresponds to a two-dimensional position in the domain, and is used to compute the density values p_{il} for each distribution i. For the third stage, we cast a ray through every pixel l of the tile. In this stage, p_{il} represents the maximum density value encountered along the ray for each distribution i, analogous to the maximum intensity projection, see Section 4.6.

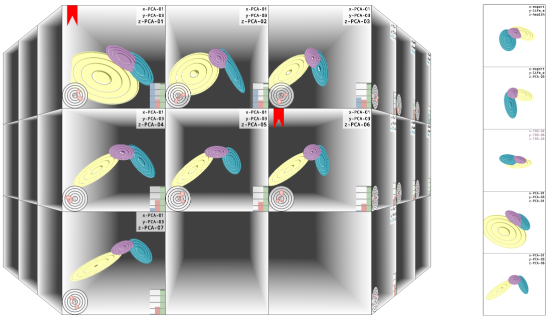
The three metrics above allow for quantitative comparisons of tiles, each corresponding to a certain basis vector choice. We express the metrics by a bar chart in the lower right corner. We chose a bar chart since it expresses quantitative values on an aligned scale, which ranks highest in effectiveness among all magnitude channels [Mun14a]. With this, the user can easily judge which metrics are most prominent. To maintain consistency, and given that the variance scores are not comparable between tiles of different facets, tiles can be reordered within a facet, but not globally.

4.4. Bookmarks

By grouping the different basis vectors into facets, the facet view supports only the comparison of basis vector choices within the same group, i.e., on the same facet. To allow the user to compare basis vector choices from different groups, we implemented bookmarks, shown in Fig. 5. Supporting task **T2**, marked tiles are pinned on the right-hand side of the screen. The camera rotates smoothly to a selected tile upon clicking on its bookmark.

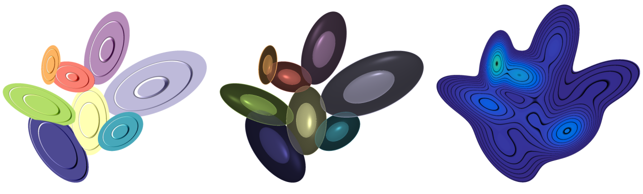
4.5. Attribution Glyph

Attribution proved useful for interpretability [BTB13, CMN*16]. To increase the interpretability of the basis vectors, which is in support of **T4**, we insert an attribution glyph in the lower left corner of each tile. See Fig. 4 (A) for a depiction of the attribution glyph. For example, consider a basis vector $\mathbf{b} = (\sqrt{0.5}, \sqrt{0.5}, 0, \ldots, 0)^{\mathrm{T}}$, which is non-zero in only its first two coefficients. This vector can be expressed as a linear combination of the canonical basis vectors $\mathbf{c}_1 = (1, 0, \ldots, 0)^{\mathrm{T}}$ and $\mathbf{c}_2 = (0, 1, 0, \ldots, 0)^{\mathrm{T}}$, i.e., $\mathbf{b} = \sqrt{0.5}\,\mathbf{c}_1 + \sqrt{0.5}\,\mathbf{c}_2$. Each canonical basis vector corresponds directly to one attribute of the data set. For a k-dimensional basis vector, we have k quantitative values to encode if the full attribution should be displayed. Several design choices are imaginable. We decided against pie charts due to the poor angle perception. Another option is the polar area chart, which is less precise when values become too small. We decided to use flower glyphs since only one item is shown in the plot at a time and those are a suitable alternative to star glyphs [vOVR22]. A flower glyph uses a radial layout for the k values. It expresses each quantitative value in terms of the size of a teardrop shape and its spatial position along the radial axis. Since eigenvectors are normalized, the value range is $[-1, 1]$, and we display the absolute value, whose range is $[0, 1]$. The glyph's segments are arranged clockwise, starting at the 12 o'clock position. The order is given by the user, e.g., an alphabetical arrangement helps locating an attribute by name. An ordered list of all names appears when hovering over the glyph, as shown in the supplemental material. The attribute name of the currently hovered glyph segment is highlighted.



Prism View with Bookmarked Tiles *Pinned Tiles*

Figure 5: *Bookmarks (red ribbons) can be added to the tiles, which adds those tiles on the right-hand side to a list of pinned tiles. This way, tiles of different facets can be compared.*



MIP View *Hull View* *DVR View*

Figure 6: *The detail view offers the three visualization techniques of [LME*23]. The MIP view conveys cluster memberships, the hull view conveys the spatial arrangement, and the DVR view helps discover new modes arising from superpositions of distributions.*

4.6. Detail View

Once all three basis vectors are chosen, the prism view transitions to the detail view. Analogous to [LME*23], we derive three views for Student t-distributions, see Fig. 6. All three detail views are embedded in 3D space. While it is conceptually imaginable to render or approximate the views differently, for example by rasterizing ellipse proxies, we follow [LME*23] and apply a ray tracing formulation for the image synthesis, since two views define the scene implicitly and the third view uses a view ray integral.

1. Maximum Intensity Projection (MIP). From all distributions, a maximum intensity projection locates the largest value along a view ray $\mathbf{p}_0 + \tau \cdot \mathbf{r}$, starting at $\mathbf{p}_0$ and traveling in unit direction $\mathbf{r}$. While this approach loses depth perception, it conveys the cluster membership of the point along the ray with the highest certainty. Upon variation of the travel distance τ along the ray, we determine analytically where the maximum is reached by requesting that the first derivative vanishes and then solving for τ:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\widehat{f}(\mathbf{p}_0 + \tau\mathbf{r}) = 0.$$

The supplemental material contains a closed-form solution that determines the depth of the maximum analytically without the need for a numerical sampling along the view ray.

2. Hull View. The hull view displays nested isocontours, which are used to convey the three-dimensional arrangement of the distributions. Isocontours with isovalue h are defined implicitly:

$$\widehat{f}(\mathbf{p}_0 + \tau\mathbf{r}) = h.$$

The supplemental material shows a quadratic closed-form solution for τ.

3. Direct Volume Rendering (DVR). Visualizing ray integrals over all distributions reveals new extrema that arise due to superpositions of distributions. For a distribution, the ray integral is

$$\int_{-\infty}^{\infty} \widehat{f}(\mathbf{p}_0 + \tau\mathbf{r})\,\mathrm{d}\tau,$$

which we integrate numerically via a Riemannian sum from near to far plane. The resulting scalar value is color-coded per pixel using the Viridis color map. In addition, isocontours are added.

In the detail view, the individual data points can be overlaid, as shown later in Figs. 9 and 12. The user has two options for the point encoding. The first encoding visualizes the cluster membership for **T5**. A pie chart encodes the probabilities of belonging to a particular cluster, as proposed by [LME*23]. Second, the points are colored according to the most likely cluster. Every point with a Mahalanobis distance to its mean greater than 2 gets a black halo, supporting **T7**. The halo thickness starts at 12.5 % and reaches up to 50 % of the radius when the distance exceeds 8. For both encodings, a range filter can be applied to discard points by maximum cluster membership or Mahalanobis distance, respectively. Both encodings and the range filter contribute to the verification of the model fit **T6**.

5. Exploration Workflow

Using Kaggle's country data set [Kok], we exemplify how a user would typically approach a data set exploration. The nine-dimensional data set includes 167 countries and has been clustered with a t-distribution mixture model. All natural axis have been scaled to a variance of one before clustering, as indicated by the blue bar in Fig. 7. The figure shows the three canonical basis vectors with the highest variance in the purple distribution. At this point, we could decide to choose these three basis vectors one after another and explore them further. This would result in an easily interpretable space since the canonical basis vectors have a direct semantic meaning. However, another option is choosing basis vectors from the per-distribution PCA of the purple cluster. This option provides the best information gain if the basis vectors with the most variance are chosen in each stage. The bar charts and the sorting function help to find these basis vectors. Here, we would expect that the three canonical basis vectors 'exports', 'health', and 'imports' are well represented during the selection process, which is confirmed by Fig. 8. The first chosen basis vector consists mainly of the canonical basis vectors 'imports' and 'exports' as shown by the attribution glyph. In the second stage, the attribution glyph reveals that the canonical basis vector 'health' is highly present. In the third stage, many more canonical basis vectors are present. As a result, the sparsity metric drops to approximately half the value of the previous two stages. In the detail view, we can filter points according to their distance to the cluster mean, as shown in Fig 9. On the right-hand side, we see a purple point near its cluster mean, which is the Bahamas. The black halo marks it as an outlier.

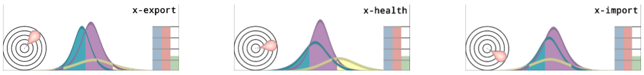
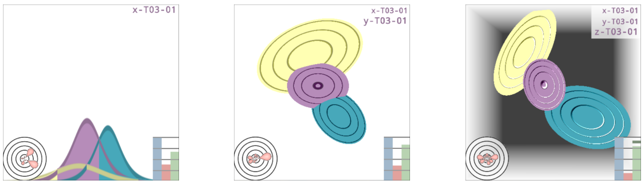


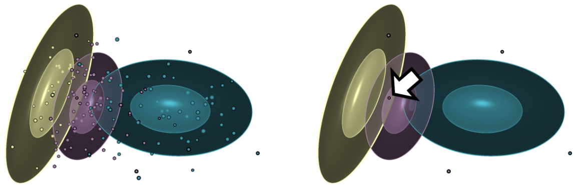
Figure 7: *Exploration entry point. Here, we see the three canonical basis vectors with the most variance in the purple distribution.*



First Stage *Second Stage* *Third Stage*

Figure 8: *The guided exploration takes the user from the first to the second, and finally to the third stage, adding one basis vector with each step. Note that in the third step, the attribution glyph shows that the third basis vector is composed of many more canonical basis vectors than the basis vectors chosen in the other two stages.*

As a second example, we examine the Shanghai ranking 2024 [Sha], which evaluates 1000 universities. The final score is the weighted sum over six categories which have been scaled to a variance of one before been clustered with a Gaussian Mixture Model. Examining the facet of the canonical basis vectors gives us first insights, see Fig 10. The order of the clusters is mostly consistent, which can be interpreted as follows: high-ranked universities perform well in all categories, while low-ranked universities perform less well overall. Distributions with a lower mean tend to represent more universities and have lower variance. Furthermore, the blue, yellow, and orange clusters have low means and barely any variance in alumni and staff awards. Interestingly, the orange distribution performs within the remaining scores much better. Especially in terms of publications, it is on par with the high-performing pur-



Hulls with all Points *Hulls with Outliers only*

Figure 9: *To detect outliers the filter distance to the mean is set to eight standard deviations or higher. The arrow marks the Bahamas, which are in the three-dimensional space close to its cluster mean.*

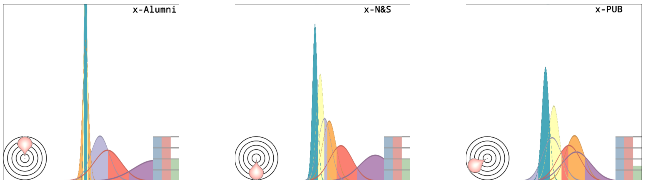


Figure 10: *The* Alumni Awards *show a low score and variance for the universities of the yellow, orange and blue cluster. The* Nature and Science publications *exhibit a clear ranking of the clusters. However, in overall* Publications *the universities of the orange cluster are en par with the top universities of the purple cluster.*

ple distribution. The first PCA tile reveals that all canonical basis vectors are relevant to obtain high variance within the data set. The per-distribution PCA facets show that every cluster has the highest variance in another canonical basis vector. To find a first basis vector, experts look for interpretable tiles in which the distributions are separable. To select a tile with high interpretability, we first consider the canonical basis vectors. Ordering the tiles according to the visibility metric helps us with this task. The "Nature and Science Paper" tile separates the distributions best. As the second basis vector, we choose the second PCA axis since it has the highest visibility on its facet and separates the distributions well. Some tiles of the second stage let us anticipate the gap between the high-performing universities represented by the purple distribution and the remaining universities. In the third stage the first PCA basis vector gives good visibility and is therefore chosen. The detail view allows us to further investigate the orange cluster. The orange distribution represents mainly Asian universities. One possible explanation is that Asia and its universities have developed rapidly in recent decades. Therefore, they had less time to win awards compared to their European and North American counterparts. In Fig. 11, we can observe the high distance between Harvard University and the other universities. Encoding the distribution probability allows the discovery of points located between two or more distributions. As Fig. 12 shows, seven universities are located between the blue and yellow distribution, and the gray and the red distribution. Only two points lie between the orange and the gray distribution, and the orange and the yellow distribution.

A third data set is investigated in the supplemental material, containing average ratings from 980 tripadvisor.com users [RSJ18].

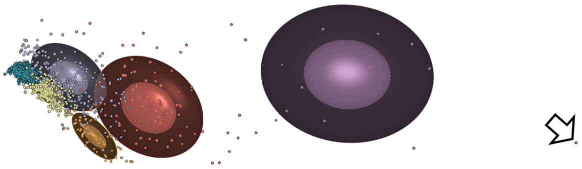


Figure 11: *The detail view shows a large distance between the purple cluster of the top universities and the remaining universities. Further, it shows a significant gap between Harvard, marked by the arrow, and the remaining top universities.*

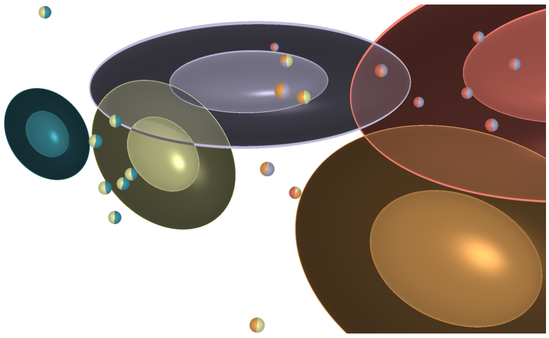


Figure 12: *Here, only data points with a maximum cluster probability less than 55 % are shown. Most edge cases can be found between the gray and the red, or the yellow and blue clusters.*

6. Evaluation

We evaluated our tool with the domain scientists who participated in the contextual inquiries (Section 3.3) to verify our approach. They regarded this tool as a novel contribution, as the study in the supplemental material shows. Additionally, we conducted a task-based evaluation, a questionnaire, and performance measurements.

6.1. Quantitative Task-based Evaluation

To assess the effectiveness of our proposed framework, we conducted an expert study with five experts, who were not involved in the development process of the tool. The experts have between two and seven years of experience in mixture models, machine learning, visualization, and/or human-computer interaction. The tasks were tailored to the requirements listed in the Requirement Analysis (Section 3.3). First, participants received files containing five clusters of dimension twelve (five weights, means, and covariance matrices). The model-based tasks included:

1 For cluster no. 1, find the three axes with the highest variance.
2 Based on cluster no. 2, which natural axes make up the axis with the largest variance?
3 Create three 1D figures in which cluster no. 3 and no. 4 are separated well.
4 Starting from cluster no. 5, find three canonical basis vectors so that the variance is large.

For the data-specific tasks, we shared five clusters of dimension twelve and 100 points within each cluster. The tasks were:

5 Find five data points that have a maximum probability of cluster membership of 80%.

6 Look at the clusters and the underlying data points. For which cluster do you think the data points do not follow a normal distribution?
7 Find the points that are at least 7 standard deviations away from the center of each cluster.

The above tasks included taking screenshots and explaining what they did to answer the questions.

First, the experts read the tasks, and we provided further explanations when necessary. Afterward, we introduced our tool using the country dataset and allowed them time for independent exploration. Then, the quantitative study for the tool began. The experts solved the tasks one by one, initiating and stopping the timer at their discretion. They required between 3:30 and 7:49 minutes to complete all tasks correctly using our framework. The maximum time recorded for a single task was 2:36 minutes. The second part of the study was conducted on the same day in the experts' typical work environments via an online meeting. At the beginning of the meeting, we sent them the tasks and dataset. All tools were permitted for solving the tasks, and all experts chose to use Python. We provided a Python function to read the data into numpy arrays to streamline the process. The experts familiarized themselves with the dataset, examining its structure (e.g., shape and storage format), and asked questions as needed. Subsequently, they solved the tasks one by one, starting and stopping the timer themselves. They were encouraged to first solve the tasks analytically, followed by a second iteration to address the visual tasks. The fastest expert completed all analytical tasks within 20:24 minutes and took an additional 28:01 minutes to solve two of the visual tasks (1 and 6). The expert who took the longest to solve the analytical tasks correctly required 49:37 minutes, but did not attempt any visual tasks. Overall, the experts required between 47:31 and 51:44 minutes to solve the tasks using their tools. On average, they took eight times longer to complete the tasks compared to using our framework. Additionally, the experts were able to create only two or fewer figures to verify their results visually when working with their tools. After the study, we asked the experts for their impressions. They highlighted three major benefits:

- the ability to gain a comprehensive overview of basis vectors,
- significant time savings achieved by using the framework, and
- a steep learning curve, enabling them to independently solve all tasks after only 20 minutes of introduction.

Exact time measurements and selected figures created by the experts are provided in the supplemental material.

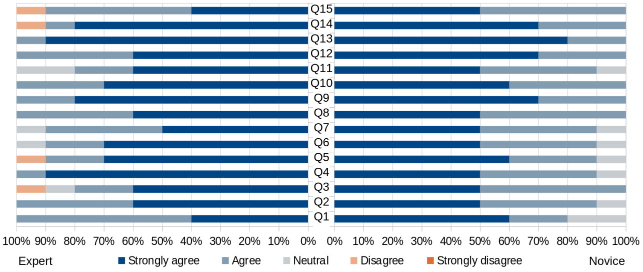


Figure 13: *Summary of the study results. Here, the individual responses of the experts (left) and non-experts (right) are shown for each question. Blue shades correspond to affirmative answers.*

6.2. Questionnaire

For our quantitative study, we used a questionnaire to assess the effectiveness of our framework. The original questions are included in the supplemental material. A total of 20 people participated in the study. According to their responses, 14 were male and six were female, and their ages ranged from 25 to 40. When asked which disciplines the participants were familiar with, multiple answers were possible: 17 said they were familiar with ML, 19 with Vis, and 16 with HCI. Six participants had one to ten years of experience, nine between ten and 19, and five between 19 and 28, with an average of 9.5 and a median of eight years of experience. We used a series of questions, each rated on a five-point Likert scale ranging from one (strongly disagree) to five (strongly agree). Additionally, participants could provide optional free-text feedback. A summary is presented in Fig. 13. The questionnaire began with questions about the first stage of the prism. Most participants (90%) strongly agreed that the prism effectively highlights facets worth exploring (e.g., canonical basis, PCA). They confirmed that the overview of basis vector choices helps identify an interesting initial basis vector for mixture models and clarifies where the probability of a specific distribution is highest. One participant noted, "It is easy to find a basis vector that separates the clusters well." For the second stage, participants agreed that the selection of a second basis vector facilitated well-separated distributions and that similar projections were easy to identify. For the third stage, participants responded positively, stating that the building metaphor effectively conveys a three-dimensional impression of multivariate distributions. In the fourth stage, where mixture models and data points are displayed together, participants were asked if the visualization helps confirm the fit between distributions and data points and identify outliers. While responses to the first question ranged from neutral to positive (some noted difficulties due to the subspace view), the second question received more favorable ratings. Next, we evaluated the effectiveness of the metric charts. Participants were asked whether the metric chart helps to identify the basis vector with the highest score, conveys performance across metrics, and helps finding similar-performing vectors for single or multiple metrics. They were also asked if reordering helps in identifying vectors with similar performance. The responses confirmed that the metric chart is valuable and effective. We then asked about the attribution of canonical basis vectors to a given vector, which participants also confirmed to be useful. Regarding bookmarks, we assessed whether they help mark interesting subspaces and allow comparisons across facets. Finally, participants provided additional feedback. One participant noted that the smooth transition between stages significantly eases the exploration process.

6.3. Performance

To assess performance, we followed a similar approach to Lawonn et al. [LME*23]. We generated three k-dimensional mixture models for $k = 6, 12, 18$. Each model consists of $k+1$ equally weighted t-distributions, with mean positions corresponding to the vertices of a k-dimensional simplex and a covariance matrix $\Sigma = \mathbf{I}$. Additionally, we sampled $10,000$ points for each mixture model. We conducted

Table 1: *Average rendering times in ms for a k-dimensional dataset with* $k+1$ *t-distributions at a resolution of* 3840×2160. *The set includes* $10,000$ *points rendered in the detail view.*

Stage (Rendering)	$k = 6$	$k = 12$	$k = 18$
1	0.60192	0.765367	0.967857
2	0.55274	1.10414	2.10762
3 (MIP)	2.75559	10.3602	33.3151
3 (Hulls)	2.45783	8.73736	22.5291
3 (DVR)	8.56939	22.6289	41.3969
Detail View (MIP)	1.88022	3.44845	5.71082
Detail View (Hulls)	1.35836	2.09915	2.72232
Detail View (DVR)	12.9403	22.8568	33.0279

Table 2: *Average loading times and standard deviation for a k-dimensional dataset with visibility metric* ($\mathrm{t_{vis}}$) *and without* (t).

Loading	k	∅ $\mathrm{t_{vis}}$ [s]	std $\mathrm{t_{vis}}$ [s]	∅ t [s]	std t [s]
Dataset	6	0.30684	0.01475	0.11833	0.00361
Dataset	12	1.49768	0.00584	0.21090	0.00621
Dataset	18	4.68488	0.00898	0.30893	0.00561
Stage 2	6	0.11384	0.01350	0.01183	0.00537
Stage 2	12	1.14037	0.03022	0.01379	0.00421
Stage 2	18	4.40502	0.00876	0.02178	0.00103
Stage 3	6	0.12927	0.01281	0.02153	0.00255
Stage 3	12	1.01967	0.00164	0.03703	0.01543
Stage 3	18	4.41620	0.00905	0.08840	0.01951
Detail View	6	-	-	0.11763	0.02381
Detail View	12	-	-	0.17403	0.01680
Detail View	18	-	-	0.22576	0.01859

two experiments using an Intel Core i7 @3.60GHz, 32 GB RAM, and an NVIDIA GeForce RTX 3070ti. In the first experiment, we measured rendering times for all stages and views as the camera rotated 360 degrees longitude around the prism center. As shown in Table 1, the tool achieves real-time performance with a minimum of 23 frames per second when displaying the third stage with DVR for $k = 18$. In all other cases, it maintains at least 30 frames per second. In the second experiment, we measured the time required to open the dataset, enter the second stage, third stage, and the detail view, each time selecting a random tile. After each run, we switched between datasets before repeating the measurements for a specific dataset. As shown in Table 2, the most time-consuming step is determining the visibility scores. To reduce loading times between stages, we allow users to deactivate this metric. For our tests, this adjustment is particularly relevant for the 18-dimensional dataset, where loading times average up to 4.4 seconds. For the other datasets, the tool demonstrates efficient response times, with a maximum average loading time of 1.1 seconds.

7. Discussion

Increasing Number of Dimensions. As the number of dimensions increases, the number of tiles per facet grows as well, resulting in smaller tiles and UI elements with less visibility. As for the attribution (flower) glyph, Onzenoodt et al. [vOVR22] assessed how accurately users could estimate the values of each segment for up to 13 dimensions. We refer to the supplemental material for additional results on data sets with varying number of dimensions.

Increasing Number of Distributions. Another challenge is the growing number of distributions. To address this challenge, we allow users to scale axes and hide distributions as needed, which is referred to as elision [Mun14b].

Choice of Metrics. Our system uses metrics, namely variance, sparsity, and visibility, to guide the user through the assembly of a projection basis. If users are not interested in finding a projection that meets those specific criteria, there is a clear risk that our system guides them away from their needs. The dependence of quality metrics on the data itself, the users, and their tasks has been a well-documented relationship [BBK*18].

Sparsity in Higher-Dimensions. As the number of dimensions increases, the sparsity metric alone becomes less descriptive, since high-dimensional non-canonical basis vectors are rarely formed from few attributes. In the supplemental material, we included an experiment on a 28-dimensional data set, where the tiles are sorted by sparsity. In practice, we recommend to use the sparsity metric in conjunction with other metrics on higher-dimensional data sets.

Further Distributions. Our system could be extended to contain further distributions. For example, the log-normal distribution, parameterized by a mean vector and covariance matrix, can be projected analogously to Eqs.(3) and (4). These projections can then be visualized either by deriving closed-form expressions or through numerical computations, as described in Section 4.6. The Cauchy distribution is inherently included in our framework as a special case of the multivariate t-distribution when $\nu = 1$.

8. Conclusion and Future Work

The visual exploration of high-dimensional, multi-variate mixture models is a challenging task. In this paper, we developed a multi-faceted view that allows the user to explore linear subspaces onto which the data distributions can be projected. Our system guides the user in a step-by-step process, adding one basis vector at a time. For each choice, several options are visualized, and quantitative quality metrics regarding retained variance, sparsity, and visibility are augmented. The interactive visualization system supports the user in various data-driven and model-driven exploration tasks. Users, including visualization and data science experts, have evaluated the system's effectiveness.

In the future, we would like to explore comparisons of multiple prism views shown side-by-side. At present, the tiles can be sorted according to three different metrics. Clustering the tiles would enhance the scalability to a higher number of tiles, leading to a hierarchical exploration of basis vector choices. Further, the rendering could potentially be accelerated by rasterizing ellipsoid proxies. At present, our tool is designed to support concrete tasks. Aiming for a more general system that supports a wider set of high-level tasks would be another interesting avenue for future work.

Acknowledgements

This work was supported by the Carl Zeiss Foundation as part of the Interactive Inference project.

References

[AdOL04] Artero A. O., de Oliveira M. C. F., Levkowitz H.: Uncovering clusters in crowded parallel coordinates visualizations. In *IEEE Symposium on Information Visualization* (2004), IEEE, pp. 81–88. doi:10.1109/INFVIS.2004.68. 2

[AWBM18] Andrews J. L., Wickins J. R., Boers N. M., McNicholas P. D.: teigen: An R package for model-based clustering and classification via the multivariate t distribution. *Journal of Statistical Software 83*, 7 (2018), 1–32. doi:10.18637/jss.v083.i07. 4

[BBK*18] Behrisch M., Blumenschein M., Kim N. W., Shao L., El-Assady M., Fuchs J., Seebacher D., Diehl A., Brandes U., Pfister H., Schreck T., Weiskopf D., Keim D. A.: Quality metrics for information visualization. *Computer Graphics Forum 37*, 3 (2018), 625–662. doi:10.1111/cgf.13446. 10

[BGG20] Bartolovic N., Gross M., Günther T.: Phase space projection of dynamical systems. *Computer Graphics Forum 39*, 3 (2020), 253–264. doi:10.1111/cgf.13978. 3

[BMH*19] Becht E., McInnes L., Healy J., Dutertre C.-A., Kwok I. W., Ng L. G., Ginhoux F., Newell E. W.: Dimensionality reduction for visualizing single-cell data using UMAP. *Nature biotechnology 37*, 1 (2019), 38–44. doi:10.1038/nbt.4314. 2

[BTB13] Broeksema B., Telea A. C., Baudel T.: Visual analysis of multi-dimensional categorical data sets. *Computer Graphics Forum 32*, 8 (2013), 158–169. doi:10.1111/cgf.12194. 6

[BTK11] Bertini E., Tatu A., Keim D.: Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2203–2212. doi:10.1109/TVCG.2011.229. 2

[CMN*16] Coimbra D. B., Martins R. M., Neves T. T., Telea A. C., Paulovich F. V.: Explaining three-dimensional dimensionality reduction plots. *Information Visualization 15*, 2 (2016), 154–172. doi:10.1177/1473871615600010. 6

[CRHC18] Chan D. M., Rao R., Huang F., Canny J. F.: t-SNE-CUDA: GPU-accelerated t-SNE and its applications to modern data. In *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)* (2018), pp. 330–338. doi:10.1109/CAHPC.2018.8645912. 2

[CvW11] Claessen J. H., van Wijk J. J.: Flexible linked axes for multivariate data visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2310–2316. doi:10.1109/TVCG.2011.201. 2

[DDH*25] Dittmann G., Ding S., Hopmans E. C., Schröter S. A., Orme A. M., Kothe E., Lange M., Gleixner G.: Bioavailable carbon additions to soil promote free-living nitrogen fixation and microbial biomass growth with N-free lipids. *Soil Biology and Biochemistry 203* (2025), 109748. doi:10.1016/j.soilbio.2025.109748. 1

[EDF08] Elmqvist N., Dragicevic P., Fekete J.-D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1539–1148. doi:10.1109/TVCG.2008.153. 2

[GHS*19] Galli E., Hartmann F. J., Schreiner B., Ingelfinger F., Arvaniti E., Diebold M., Mrdjen D., van der Meer F., Krieg C., Nimer F. A., et al.: GM-CSF and CXCR4 define a T helper cell signature in multiple sclerosis. *Nature medicine 25*, 8 (2019), 1290–1300. doi:10.1038/s41591-019-0521-4. 2

[GLP*24] Giesen J., Lucas P., Pfeiffer L., Schmalwasser L., Lawonn K.: The whole and its parts: Visualizing Gaussian mixture models. *Visual Informatics 8*, 2 (2024), 67–79. doi:10.1016/j.visinf.2024.04.005. 3

[Hot33] Hotelling H.: Analysis of a complex of statistical variables into principal components. *Journal of educational psychology 24*, 6 (1933), 417. doi:10.1037/h0071325. 2

[JCC*11] Joia P., Coimbra D., Cuminato J. A., Paulovich F. V., Nonato L. G.: Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2563–2571. doi:10.1109/TVCG.2011.220. 2

[Kaw08] Kawabata T.: Multiple subunit fitting into a low-resolution density map of a macromolecular complex using a Gaussian mixture model. *Biophysical Journal 95* (2008), 4643–4658. 1

[Kok] Kokkula R.: Unsupervised learning on country data. https://www.kaggle.com/datasets/rohan0301/unsupervised-learning-on-country-data. Accessed: 2023-11-05. 7

[LHPW16] Li L., Hansman R., Palacios R., Welsch R.: Anomaly detection via a Gaussian mixture model for flight operation and safety monitoring. *Transportation Research Part C: Emerging Technologies 64* (2016), 45–57. 1

[LME*23] Lawonn K., Meuschke M., Eulzer P., Mitterreiter M., Giesen J., Günther T.: GRay: Ray casting for visualization and interactive data exploration of Gaussian mixture models. *IEEE Transactions on Visualization and Computer Graphics 29* (1 2023), 526–536. doi:10.1109/TVCG.2022.3209374. 2, 3, 4, 6, 7, 9

[LMW*17] Liu S., Maljovec D., Wang B., Bremer P., Pascucci V.: Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics 23*, 3 (2017), 1249–1268. doi:10.1109/TVCG.2016.2640960. 2

[LT13] Lehmann D. J., Theisel H.: Orthographic star coordinates. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2615–2624. doi:10.1109/TVCG.2013.182. 2

[LT15] Lehmann D. J., Theisel H.: Optimal sets of projections of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 609–618. doi:10.1109/TVCG.2015.2467132. 2

[Mat14] Mathieu Jacomy: iwanthue. Website, 2014. Called Mar. 03, 2025 from: https://medialab.github.io/iwanthue/. 5

[MG13] Mayorga A., Gleicher M.: Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics 19*, 9 (2013), 1526–1538. doi:10.1109/TVCG.2013.65. 2

[MHM18] McInnes L., Healy J., Melville J.: UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018). URL: https://arxiv.org/abs/1802.03426. 2

[Mun14a] Munzner T.: *Visualization Analysis and Design*. CRC press, 2014. 2, 6

[Mun14b] Munzner T.: *Visualization analysis and design*. CRC press, 2014. 10

[OHJ*11] Oesterling P., Heine C., Janicke H., Scheuermann G., Heyer G.: Visualization of high-dimensional point clouds using their density distribution's topology. *IEEE Transactions on Visualization and Computer Graphics 17*, 11 (2011), 1547–1559. doi:10.1109/TVCG.2011.27. 2

[Par22] Parsons P.: Understanding data visualization design practice. *IEEE Transactions on Visualization and Computer Graphics 28*, 1 (2022), 665–675. doi:10.1109/TVCG.2021.3114959. 3

[PEP*11] Paulovich F. V., Eler D. M., Poco J., Botha C. P., Minghim R., Nonato L. G.: Piece wise Laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum 30*, 3 (2011), 1091–1100. doi:10.1111/j.1467-8659.2011.01958.x. 2

[PTM*20] Pezzotti N., Thijssen J., Mordvintsev A., Höllt T., Van Lew B., Lelieveldt B. P., Eisemann E., Vilanova A.: GPGPU linear complexity t-SNE optimization. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 1172–1181. doi:10.1109/TVCG.2019.2934307. 2

[RSJ18] Renjith D. S., Sreekumar A., Jathavedan M.: Evaluation of partitioning clustering algorithms for processing social media data in tourism domain. *2018 IEEE Recent Advances in Intelligent Computational Systems (RAICS)* (2018), 127–131. URL: https://api.semanticscholar.org/CorpusID:59619597. 8

[Sha] ShanghaiRanking: Top universities ranking 2024. https://www.kaggle.com/datasets/computingvictor/2024-academic-ranking-of-world-universities. Accessed: 2024-10-03. 7

[SNLH09] Sips M., Neubert B., Lewis J. P., Hanrahan P.: Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum 28*, 3 (2009), 831–838. doi:10.1111/j.1467-8659.2009.01467.x. 2

[SS04] Seo J., Shneiderman B.: A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *Proceedings of IEEE Symposium on Information Visualization* (2004), pp. 65–72. doi:10.1109/INFVIS.2004.3. 2

[SSB09] Shin M.-S., Sekora M., Byun Y.-I.: Detecting variability in massive astronomical time-series data. i: application of an infinite Gaussian mixture model. *Monthly Notices of the Royal Astronomical Society 400*, 4 (2009), 1897–1910. 1

[SW09] Sanftmann H., Weiskopf D.: Illuminated 3D scatterplots. *Computer Graphics Forum 28*, 3 (2009), 751–758. doi:10.1111/j.1467-8659.2009.01477.x. 2

[TAE*09] Tatu A., Albuquerque G., Eisemann M., Schneidewind J., Theisel H., Magnor M., Keim D.: Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST)* (Oct 2009), pp. 59–66. doi:10.1109/VAST.2009.5332628. 2

[VdMH08] Van der Maaten L., Hinton G.: Visualizing data using t-SNE. *Journal of machine learning research 9*, 11 (2008). 2

[vOVR22] van Onzenoodt C., Vázquez P.-P., Ropinski T.: Out of the plane: Flower vs. star glyphs to support high-dimensional exploration in two-dimensional embeddings. *IEEE Transactions on Visualization and Computer Graphics* (2022). doi:10.1109/TVCG.2022.3216919. 6, 10

[War08] Ward M. O.: *Multivariate Data Glyphs: Principles and Practice*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 179–198. doi:10.1007/978-3-540-33037-0_8. 2

[WHK90] Wixon D., Holtzblatt K., Knox S.: Contextual design: an emergent view of system design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1990), CHI '90, Association for Computing Machinery, p. 329–336. doi:10.1145/97243.97304. 4

[WSPVJ11] Wang B., Summa B., Pascucci V., Vejdemo-Johansson M.: Branching and circular features in high dimensional data. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 1902–1911. doi:10.1109/TVCG.2011.177. 2

[WVJ16] Wattenberg M., Viégas F., Johnson I.: How to use t-SNE effectively. *Distill* (2016). doi:10.23915/distill.00002. 2

[ZBLB24] Zhang Z. T., Buchner F., Liu Y., Butz A.: You can only verify when you know the answer: Feature-based explanations reduce overreliance on AI for easy decisions, but not for hard ones. In *Proceedings of Mensch Und Computer 2024* (New York, NY, USA, 2024), MuC '24, Association for Computing Machinery, p. 156–170. doi:10.1145/3670653.3670660. 3

[ZGL] Zahoransky B., Günther T., Lawonn K.: PrismBreak. https://github.com/BrianZah/PrismBreak. available after publication. 2