

PrismBreak: Exploration of Multi-Dimensional Mixture Models

Supplemental Material

B. Zahoransky¹ , T. Günther²  and K. Lawonn¹ 

¹Friedrich Schiller University Jena, Germany

²Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

1. Maximum Intensity Projection

To locate the point along the view ray at which the distribution reaches its maximum, we differentiate with respect to τ and solve for the root:

$$\frac{d}{d\tau} \hat{f}(\mathbf{p}_0 + \tau \mathbf{v} \mathbf{r}) = \quad (1)$$

$$= \frac{d}{d\tau} \frac{\Gamma\left(\frac{v+k}{2}\right)}{\Gamma\left(\frac{v}{2}\right) \sqrt{(v\pi)^k \det(\hat{\Sigma})}} \quad (2)$$

$$\left(1 + \frac{1}{v} (\mathbf{p}_0 + \tau \mathbf{r} - \hat{\mu})^T \hat{\Sigma}^{-1} (\mathbf{p}_0 + \tau \mathbf{r} - \hat{\mu})\right)^{-\frac{v+k}{2}} \quad (3)$$

$$= \frac{\Gamma\left(\frac{v+k}{2}\right)}{\Gamma\left(\frac{v}{2}\right) \sqrt{(v\pi)^k \det(\hat{\Sigma})}} \quad (4)$$

$$- \frac{v+k}{2} \left(1 + \frac{1}{v} (\mathbf{p}_0 + \tau \mathbf{r} - \hat{\mu})^T \hat{\Sigma}^{-1} (\mathbf{p}_0 + \tau \mathbf{r} - \hat{\mu})\right)^{-\frac{v+k-1}{2}} \quad (5)$$

$$\left(\frac{1}{v} (2\mathbf{r}^T \hat{\Sigma}^{-1} (\mathbf{p}_0 - \hat{\mu}) + 2\tau \mathbf{r}^T \hat{\Sigma}^{-1} \mathbf{r})\right) \quad (6)$$

Inserting the t-density into Eq. (1) gives Eqs. (2)–(3). The application of the chain rule leads to Eqs. (4)–(6). The product vanishes when Eq. (6) vanishes to zero, which is a linear function in the distance τ . Rearranging for τ gives a closed-form expression for the distance at which the distribution reaches its maximum:

$$\tau = \frac{\mathbf{r}^T \hat{\Sigma}^{-1} (\hat{\mu} - \mathbf{p}_0)}{\mathbf{r}^T \hat{\Sigma}^{-1} \mathbf{r}} \quad (7)$$

2. Hull Surface

Hull surfaces depict isocontours

$$\hat{f}(\mathbf{p}_0 + \tau \mathbf{r}) = h \quad (8)$$

for a given isovalue h . Rearranging Eq. (8) gives:

$$(\mathbf{p}_0 + \tau \mathbf{r} - \hat{\mu})^T \hat{\Sigma}^{-1} (\mathbf{p}_0 + \tau \mathbf{r} - \hat{\mu}) = R \quad (9)$$

with

$$R = v \left(\left(\frac{h \Gamma\left(\frac{v}{2}\right) \sqrt{(v\pi)^k \det(\hat{\Sigma})}}{\Gamma\left(\frac{v+k}{2}\right)} \right)^{-\frac{2}{v+k}} - 1 \right) \quad (10)$$

Eq. (9) is quadratic in τ

$$\tau^2 a + \tau b + c = 0 \quad (11)$$

and hence there is a closed-form solution for the entry and exit distance of the ray with the coefficients:

$$a = \mathbf{r}^T \hat{\Sigma}^{-1} \mathbf{r} \quad (12)$$

$$b = 2\mathbf{r}^T \hat{\Sigma}^{-1} \mathbf{p}_0 - 2\mathbf{r}^T \hat{\Sigma}^{-1} \hat{\mu} \quad (13)$$

$$c = \hat{\mu}^T \hat{\Sigma}^{-1} \hat{\mu} - 2\mathbf{p}_0^T \hat{\Sigma}^{-1} \hat{\mu} + \mathbf{p}_0^T \hat{\Sigma}^{-1} \mathbf{p}_0 - R \quad (14)$$

Note that solutions are imaginary if the ray does not hit the isocontour.

3. Implementation Details

The following section elaborates on implementing the shaders used to efficiently render the prism view. Fig. 1 gives an overview of the rendering pipeline.

Input Data. The input to our algorithm is a mixture model as in Eq. (2) of the main paper with given distribution parameters, i.e., means and covariances. If the user provides points only, we use existing packages to fit distributions [AWBM18]. We store the number of distributions n and the parameters μ_i, Σ_i in a uniform buffer object on the GPU, allowing efficient access from all shader stages. Further, an index buffer is created on the CPU side, which stores the order in which the tiles are displayed on the facets. When the user orders the tiles by a different metric, the index buffer content is updated accordingly.

Tile Transformation. The geometric layout of the prism view is determined on the CPU side by filling a static vertex buffer. The vertex buffer contains two triangles for every tile of each facet. This way, the primitive ID can be used to identify the ID of the tile uniquely. The triangle vertices are ordered such that the first

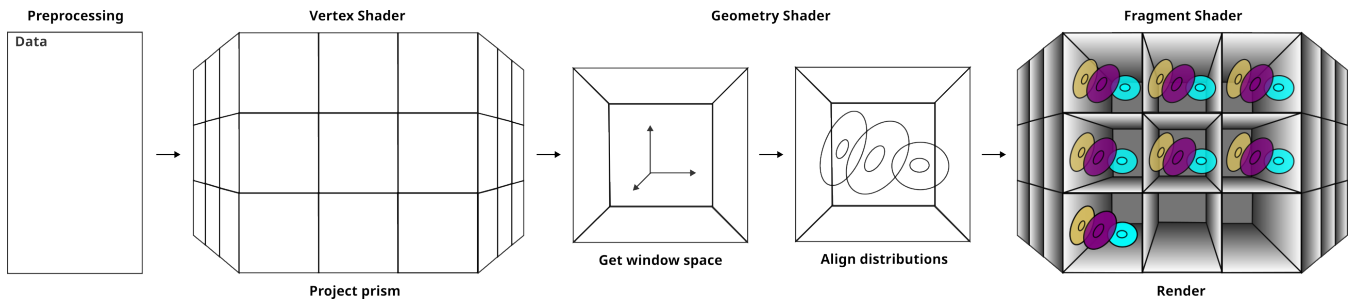


Figure 1: Illustration of the rendering pipeline. Input to our method is given mixture models or individual data points to which distributions are fitted first. The prism view is formed by a quad mesh with one quad per tile, and where each quad is represented on the GPU with two triangles. From the primitive index, the tile can be identified. The parameters of the mixture model distributions (means and covariances) are passed via uniform buffer objects. The per-tile transformation from world space to tile space is calculated to place the distributions inside the tiles. Lastly, the fragment shader performs interior mapping to calculate the intersections with the walls, and performs analytic ray casting against the transformed distributions.

two vertices constitute the tile diagonal. Its midpoint identifies the center of the tile. Since each tile has its designated location on the facet, the distributions can be read from the uniform buffer object, and their coordinate system can be transformed to the tile's respective location.

Shading. The fragment shader receives the identifier of the tile shown by a given pixel from the previous shader stages. A ray is then cast analytically against all distributions, which have been transformed to the respective tile coordinates beforehand. The third layer of the prism view conveys a 3D impression by showing walls in each cell's interior. If a ray does not intersect with a distribution, we calculate the intersection with an interior wall using the interior mapping algorithm [vD08]. To improve the depth perception, shading is calculated on the interior walls. All calculations are lightweight, and the frame rate of the algorithm is very high, as shown in Section 7.4 of the main paper.

4. Glyph Tooltip

In Fig. 2 we see a tooltip that appears when the user hovers over the attribution glyph of a tile.

5. Travel Reviews

The third investigated data set contains average ratings from 980 tripadvisor.com users [RSJ18] and was clustered with a Gaussian mixture model. All natural axis were normalized before clustering. The provided average ratings cover ten categories. We start our analysis on the facet of canonical basis vectors on the first stage. Four distributions can be seen: The users represented by the purple one review parks, resorts, and juice bars better than the others but often give religious places bad reviews, as seen in Fig. 3. The users of the blue distribution seem to be counter-reviewing religious places as good while giving bad reviews for parks and resorts. The yellow distribution evaluates restaurants much better than the remaining users, as seen in Fig. 4 left. The mean of violet distribution lies often centered, thus those users appear to have no preferences. To separate the distribution from the remaining distributions, we choose as the first axis the canonical basis vector for

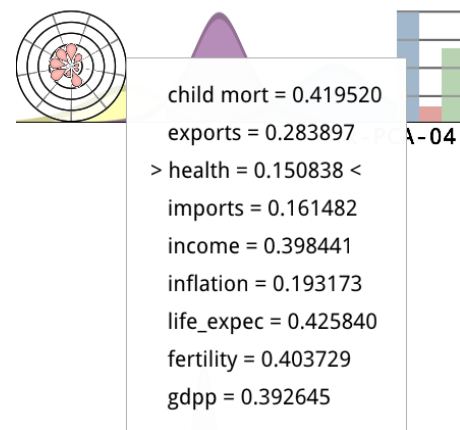


Figure 2: When the user hovers over the glyph, a tooltip pops up showing the attributes and corresponding values. The attribute the mouse points at is highlighted.

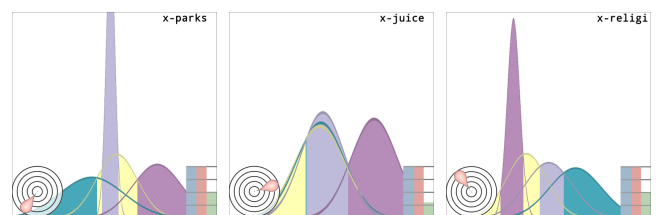


Figure 3: Here, we observe three canonical basis vectors, with the blue and purple distributions located on opposite sides within each tile.

restaurant reviews. Afterwards, we are interested in getting the best information gain for the yellow distribution. Thus, we choose the first tiles from the facet of the yellow distribution in each stage, as seen in Fig. 4. In the second, we discover that this tile is highly influenced by dance clubs. The first per-distribution PCA basis vector of the last stage is composed of many attributes. Examining the yellow distribution in the detail view reveals that even this cluster

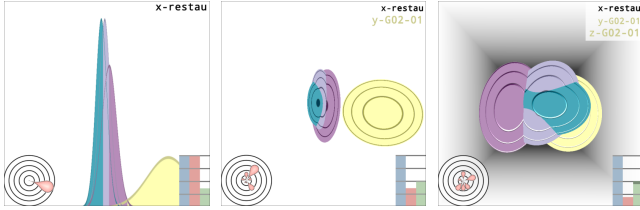


Figure 4: The guided exploration leads the user through three stages, incrementally adding one basis vector at each step. Notably, in the third stage, the attribution glyph reveals that the third basis vector is composed of significantly more canonical basis vectors compared to those selected in the earlier stages.

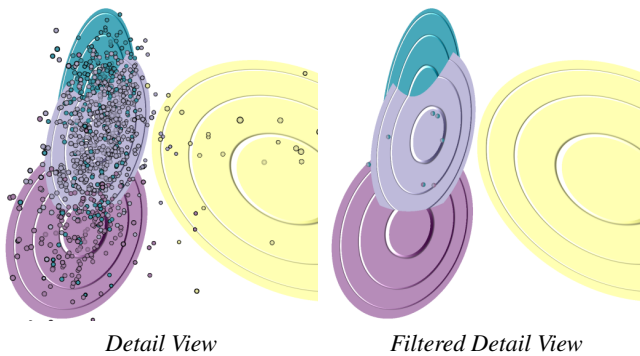


Figure 5: In the unfiltered detail of this subspace, the three clusters on the left and their points exhibit significant overlap. However, by applying the filter options, it becomes clear that the clusters are well separated in high-dimensional space. Only 7 points were identified with a maximum cluster probability of less than 55 %.

has a low average rating of only 1.7 for restaurants. But the view also raises questions if our Gaussian model was fitted well since the points of the other clusters are really crowded, as seen in Fig. 5 left. A way to verify the fitting is to experiment with different subspaces. Our tool offers, however, a more efficient way. Using the point filter reveals that only 36 points have a derivation higher than five, and only 21 points have a maximum cluster probability of less than 65 %. If we lower this value to 55 % there are only 7 points left. This allows us to conclude that the clusters are clearly separated in the high-dimensional space, but this subspace is only suited to separate the yellow distribution from the other clusters.

6. Expert Study

The expert study was conducted with the domain scientists who participated in the contextual inquiries described in the *Requirement Analysis* (Section 3). Expert D has 5 years of experience in data exploration, clustering, and analysis. Expert M has 30 years of experience in machine learning and mixture models. Finally, expert V has 12 years of experience in visualization. During the evaluation, we followed the think-aloud protocol and encouraged participants to verbally communicate their thoughts. We took note of everything they said. First, we explained the idea of building their subspace by introducing a tractable data set of four dimensions and three distributions. We introduced the prism and explained the con-

cepts of building their basis and adding bookmarks. All experts commented that these concepts were novel and made it easy to explore the data. When asked about their usual tools, they mentioned Python, R, and scatterplot matrices for 2D canonical subspaces. While some used PCA for dimensionality reduction, they noted that interpretability is challenging. M said, “The glyph acts as a signpost in high-dimensional space, describing the direction I am looking.” Besides the prism, we also showed the experts the different visualization techniques (MIP, Hull, DVR) by Lawonn et al. [LME*23]. All experts agreed on their usefulness in the respective task at hand. Expert D stated: “The DVR view beautifully demonstrates that a mixture of t-distributions is not a t-distribution itself.” Expert D sent us their own data set in advance, which they were already familiar with. Interestingly, the expert could detect features that they were unaware of before. D said, “This tool allowed me to explore the data more intuitively than before, and the filter function allowed me to spot outliers that I was not aware of.” Finally, all experts considered this tool a novel contribution that clearly fills a gap in data exploration and analysis. Experts D and M asked for the availability of source code, which we will publish after acceptance.

7. Quantitative Task-based Evaluation

Table 1 summarizes the time needed by the experts for each task. Figs. 6, 7, 8, 9, 10 and 11 show figures created by our experts during the Quantitative Task-based Evaluation.

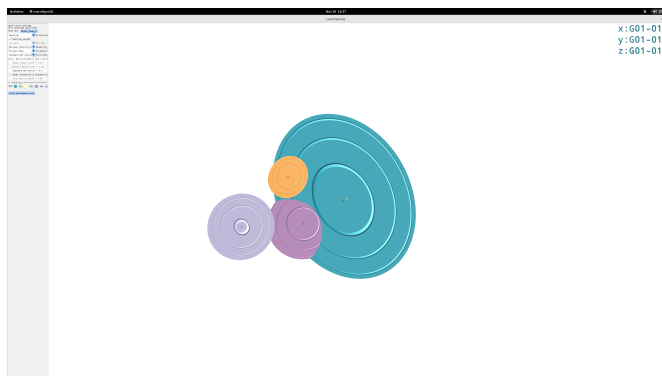
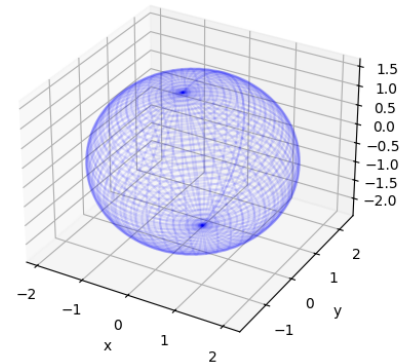
8. Questionnaire

In the following, the original questionnaire the participants received is shown.

1. The overview in the first stage of the prism view helps to identify a facet (e.g., canonical basis, PCA, etc.) that is worth exploring further.
2. In the first stage of the prism view, the overview of basis vector choices on a selected facet helps to identify an interesting first basis vector for the mixture models.
3. The visualization helps to understand where the probability of a certain distribution is highest.
4. The second stage helps to identify a second basic vector that leads to well-separable distributions.
5. Similar projections are easy to identify.
6. The building metaphor conveys a three-dimensional impression of the multi-variate distributions.
7. Visualizing the data points helps to confirm whether the distribution fits the data points well.
8. Potentially interesting data points can be identified (e.g., outliers).
9. The metric chart allows for finding the basis vector with highest score for any given metric.
10. The metric chart conveys well if a basic vector performs well on one or multiple metrics.
11. The metric chart allows for finding basic vectors that perform similarly across individual or multiple metrics.
12. The reordering helps in finding basis vectors that perform similarly according to a given metric.
13. The basis attribution conveys which canonical basis vectors contribute to a given basis vector.

Table 1: The table presents the time (in minutes) taken by each expert to complete the tasks in the *Quantitative Task-based Evaluation*.

User Id		1	2	3	4	5	6	7	overall
1	Tool	01:13	00:44	01:00	01:10	00:50	00:20	00:40	05:57
	analytical	10:00	03:38	07:26	01:04	10:14		05:04	37:26
	visual	14:18							14:18
2	Tool	02:20	00:26	01:06	00:33	00:35	02:36	00:13	07:49
	analytical	05:40	01:40	08:16	01:12	12:06	11:45	08:29	49:08
	visual								00:00
3	Tool	00:40	01:03	00:55	00:58	01:29	00:27	00:11	05:43
	analytical	04:54	01:26	05:13	03:15	15:04		10:17	40:09
	visual			09:28					09:28
4	Tool	00:30	00:38	01:31	01:39	00:41	01:47	00:13	06:59
	analytical	01:25	03:28	01:53	05:57	09:01	07:24	02:36	31:44
	visual			12:11			03:36		15:47
5	Tool	00:20	00:12	00:46	00:47	00:53	00:20	00:12	03:30
	analytical	01:46	00:42	04:40	00:57	02:57	08:04	01:18	20:24
	visual	24:20					03:41		28:01

*Our Tool**Expert Solution***Figure 6:** Task 1: For cluster no. 1, find the three basis vectors with the highest variance. Create a 3D figure where those vectors are the three axis.

14. Setting bookmarks is useful for marking interesting subspaces.
15. The overview of bookmarked basis vectors enables the comparison of basis vectors that have been selected on different facets.

9. Full Performance Measures

In Section 6 of the main paper, we discussed the rendering times of our system at all stages. Table 2 further lists the standard deviations of the time measurements. In agreement with the observations in the main paper, we found that the system reached real-time frame rates in all settings.

10. Visibility Metric

Fig. 12 illustrates a step in exploring the Shanghai ranking dataset. In the first stage, we selected the 'Nature and Science Publications'

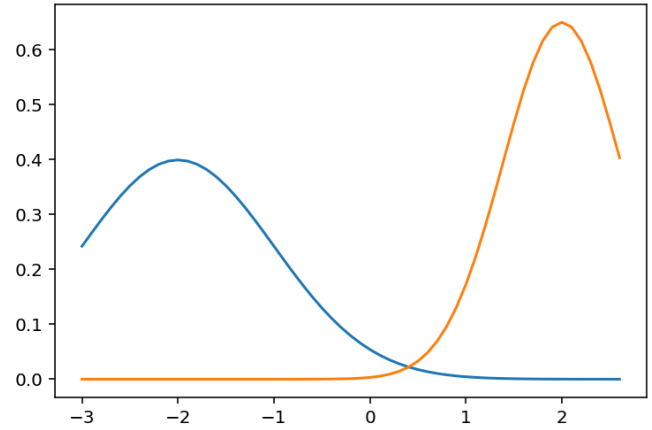
tile. The second stage shows the tiles ordered by the visibility metric. In the top-ranked tile, located in the upper left corner, the distributions overlap significantly less than in the other tiles. In contrast, the last two tiles exhibit much higher overlap.

11. Sparsity Metric

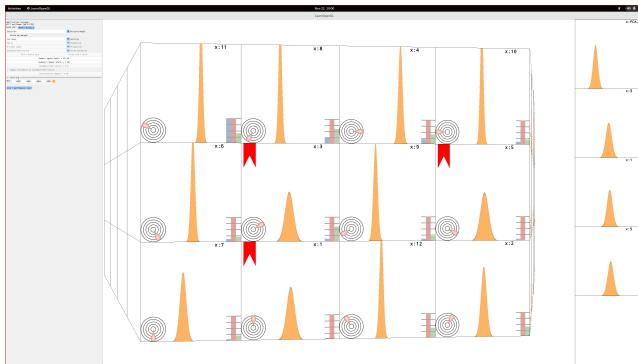
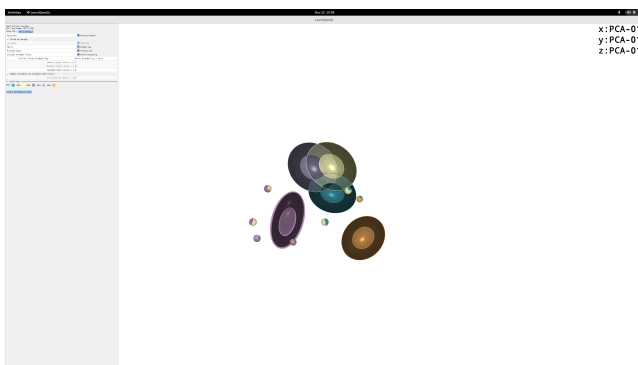
The 'Urban Air Quality and Health Impact Analysis' [Abd] dataset contains 1,000 instances across 46 dimensions. However, after excluding non-continuous attributes, only 28 dimensions remain. The PCA facet from the first stage is shown in Fig. 13, with tiles ordered by their sparsity scores. In the first and second tiles, we can observe peaks in the attribution graph. Hovering over the graph reveals that the 'feels like maximum temperature' contributes about half to the first basis vector, while the actual temperature accounts for a quarter. For the second basis vector, sunrise and sunset each contribute approximately 40%. Despite the existence of vectors with



Our Tool



Expert Solution

Figure 7: Task 3: Create three 1D figures in which cluster no. 3 and no. 4 are well separated.**Figure 8:** Task 4: Starting from cluster no. 5, find three canonical basis vectors so that the variance is large.**Figure 9:** Task 5: Find five data points that have a maximum probability of cluster membership of 80%. Create a figure demonstrating that these points lie between the clusters.**Table 2:** Average rendering times and standard deviations 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	$\varnothing t$ [ms]	std t [ms]
1	6	0.60192	0.0776197
1	12	0.765367	0.098181
1	18	0.967857	0.108132
2	6	0.55274	0.077256
2	12	1.10414	0.12213
2	18	2.10762	0.16661
3 (MIP)	6	2.75559	0.222025
3 (MIP)	12	10.3602	0.227028
3 (MIP)	18	33.3151	0.398709
3 (Hulls)	6	2.45783	0.243421
3 (Hulls)	12	8.73736	0.604935
3 (Hulls)	18	22.5291	3.20841
3 (DVR)	6	8.56939	0.420942
3 (DVR)	12	22.6289	0.476716
3 (DVR)	18	41.3969	1.90924
Detail View (MIP)	6	1.88022	0.176672
Detail View (MIP)	12	3.44845	0.18506
Detail View (MIP)	18	5.71082	0.255651
Detail View (Hulls)	6	1.35836	0.194521
Detail View (Hulls)	12	2.09915	0.214887
Detail View (Hulls)	18	2.72232	0.218801
Detail View (DVR)	6	12.9403	0.291809
Detail View (DVR)	12	22.8568	0.646743
Detail View (DVR)	18	33.0279	0.128252

high sparsity for this dataset, a user would likely never select these tiles. The simple reason is that the data shows little variance for these attributes—'feels like maximum temperature,' sunrise, and sunset times are nearly identical across all instances.

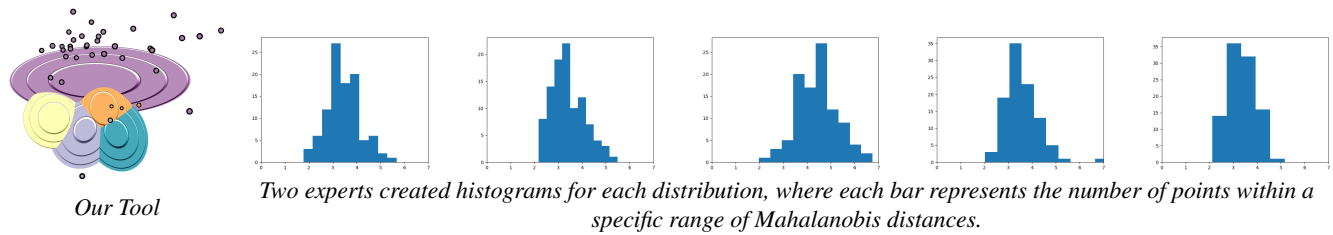


Figure 10: Task 6: Look at the clusters and the underlying data points. For which cluster do you think the data points do not follow a /revisednormal distribution? They had to identify the purple distribution, which is represented by the histogram in the middle. Notably, this histogram exhibits generally higher Mahalanobis distances.

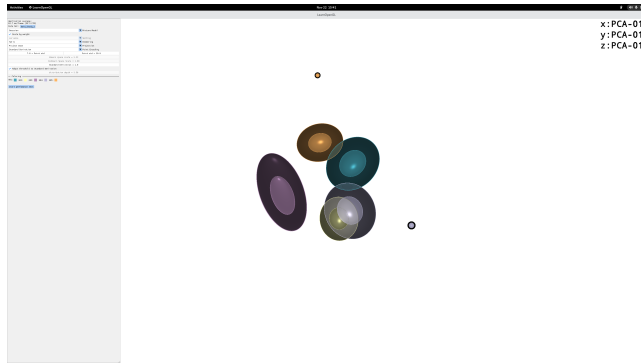


Figure 11: Task 7: Find the points that are at least 7 standard deviations away from the center of each cluster. Display those points in a figure.

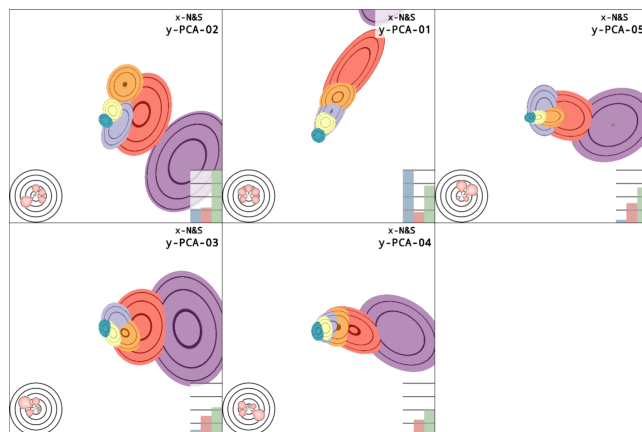


Figure 12: The second stage of the prism view sorted according to the visibility score.

References

- [Abd] ABDULLAH M.: Urban air quality and health impact analysis. <https://www.kaggle.com/datasets/abdullah0a/urban-air-quality-and-health-impact-dataset>. called. 4
- [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. 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. 3
- [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>. 2
- [vD08] VAN DONGEN J.: Interior mapping. In *CGI 2008 Conference Proceedings* (2008). 2

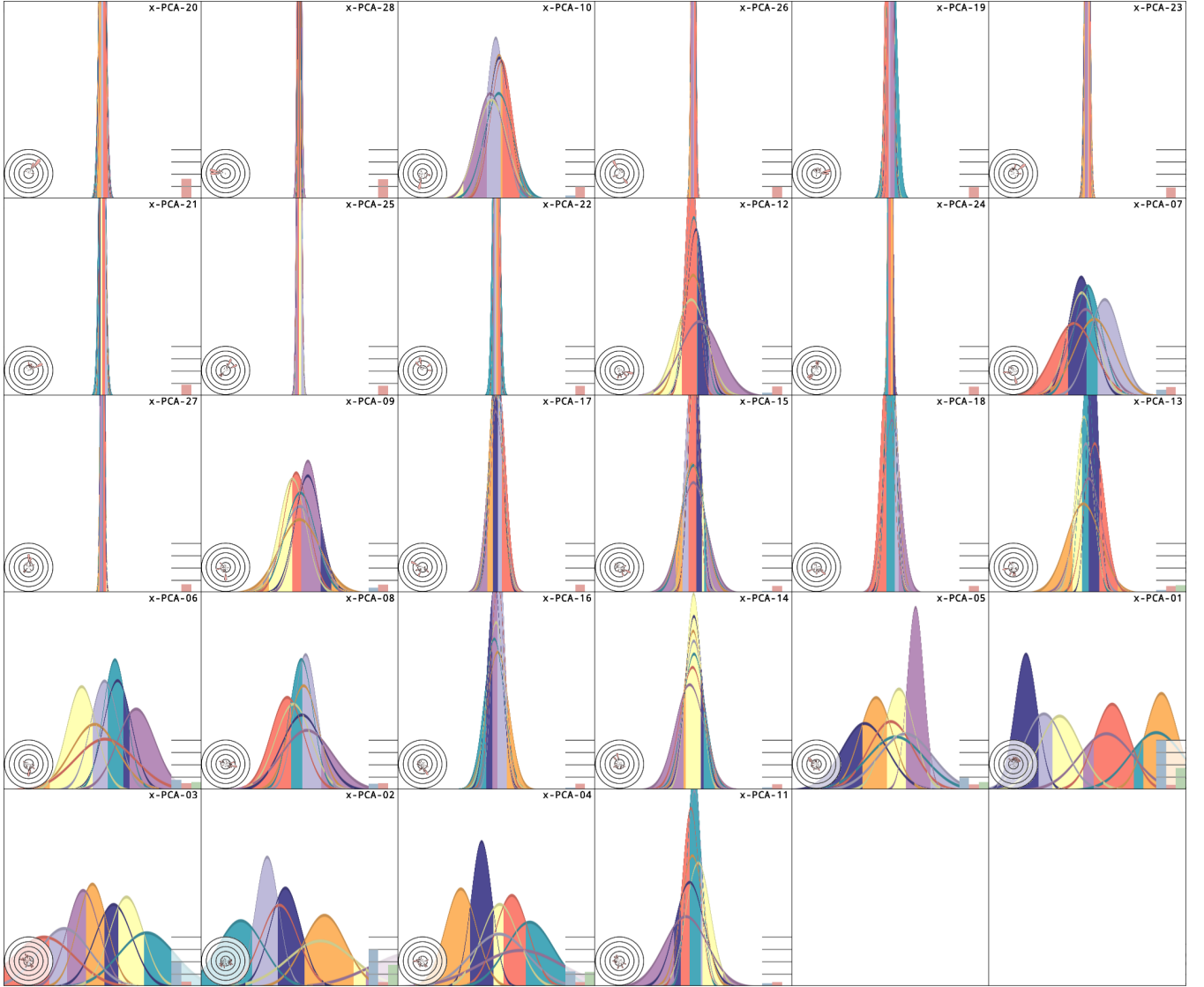


Figure 13: The PCA facet from the first stage of the prism for a 28-dimensional dataset. The tiles are ordered according to their sparsity score. A user would likely avoid selecting the highest-ranked basis vectors.