Feature Wind Map: Continuous Feature Sensitivity Visualization for Dimension Reduction

Category: Research

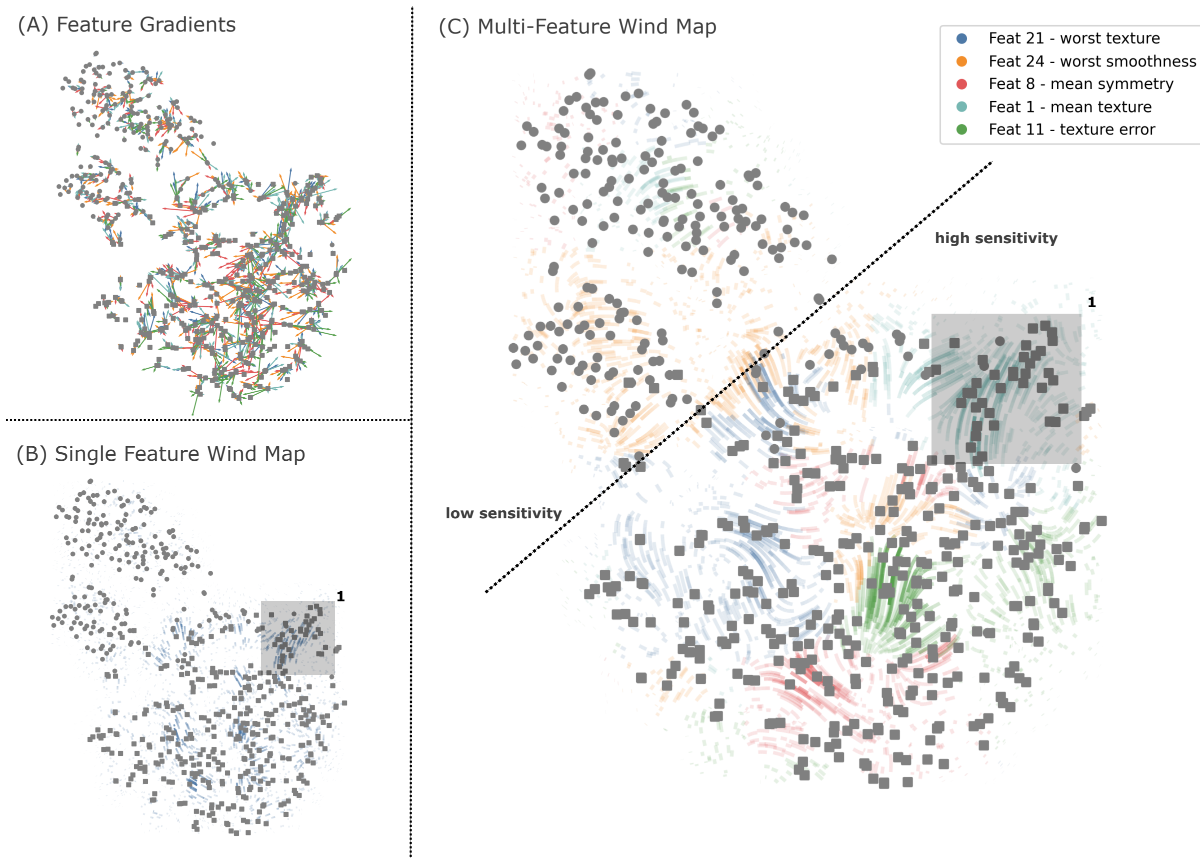

Figure 1: Feature Wind Map applied to breast cancer dataset. (A) shows one natural visual encoding of feature gradients: vectors; (B) shows the wind map for a single feature *worst texture*; (C) is the wind map for top five features sorted by the overall sensitivity across the dataset.

Abstract

Dimension reductions(DR) suffer from a well-known lack of interpretability of the projection space relative to the input features. Feature gradients help explain DR plots by capturing how small perturbations in the original features would shift each point's position in the DR projections. The natural visual encoding of these gradients is arrows on the points, where each arrow represent a gradient. However, when gradients from multiple features overlap densely on individual points, they obscure global flow patterns and make it difficult to discern coherent directions or magnitude variations. In this paper, we propose Feature Wind Map, a dynamic visualization that aggregates gradients into a smooth and continuous flow field. Feature Wind Map generates an aggregate vector field from the gradients and uses particle advection to illustrate the flow of points in this vector field. This reveals coherent direction patterns and magnitudes of the feature gradients that demonstrate global sensitivities of projected points, without overwhelming DR plots with visual clutter. We evaluate our method through case studies on synthetic and real world datasets. Our results indicate that Feature Wind Map captures which features steer the embedding, where their sensitivity switches, and how strongly each region responds.

Index Terms: Dimension Reduction, Sensitivity analysis, Visual annotation.

1 Introduction

Dimensionality reduction (DR) techniques, such as t-SNE [27], UMAP [14], and PCA [17], are widely used to project high-dimensional data into low-dimensional space. While these projections can reveal clusters, trends, and outliers, they provide limited insights into how individual features affect the spatial arrangement of projected data points [15]. Without additional visualization to reveal feature behaviors in the projection, understanding the relationship between input features and observed patterns is challenging.

Current methods for visualizing feature behavior in DR spaces typically rely on static techniques, such as coloring points by feature values, shading regions by feature influence, or overlaying discrete vectors to indicate directional trend [4, 6, 7, 12, 16, 24]. Although these approaches provide a general sense of how features relate to the projection in simple settings, they struggle to capture the dynamic and spatially varying nature of feature influence. In large datasets with complex feature interactions, static visual encodings can quickly lead to visual clutter, making it difficult to intuitively understand how different features shape the embedding structure. Moreover, users often must manually explore one feature at a time, which can obscure multi-feature inspection [7].

To address these challenges, we introduce **Feature Wind Map**, a dynamic, animated visualization that reveals feature sensitivity across 2D projection spaces (see Fig. 1C). Feature sensitivity shows how strongly small changes in an input feature affect a point's position in the projection, and it allows us to directly characterize how feature influence varies across space.

Inspired by meteorological wind maps that depict the invisible flow field [8, 18, 19], Feature Wind Map leverages dynamic motion perception to enhance the recognition of complex spatial patterns [1, 12, 26]. Our method computes feature gradients directly with respect to the projected positions, interpolates these gradients into smooth vector fields, and advects particles across the field to represent feature-driven "flows" through motion and color. The resulting wind field aggregates the sensitivities across all input features or a selected subset, enabling overall trends or feature-specific sensitivity. In contrast to approximated or discrete vector plots [16], our gradient-based framework provides a direct, fine-grained, and continuous depiction of feature sensitivities. This allows users to simultaneously perceive the direction, magnitude, and dominant influencing feature at each location, aligning naturally with human perceptual abilities and reducing visual clutter. The main contributions of this paper are:

- **Feature Wind Map**, a dynamic visualization using motion and color to reveal feature sensitivity in 2D projections. It combines:
  - A transformation of the Jacobian matrix of a DR mapping into a smooth vector field;
  - Particle advection to animate global direction and magnitude of feature sensitivities, highlighting local dominant features;
- Two **case studies** on synthetic and real-world datasets to demonstrate the interpretability and usage of Feature Wind Map.

2 Related Work

Feature Sensitivity for Explaining DR Projections Feature sensitivity analysis is a fundamental approach for understanding the influence of input variables on a model's output [20]. Gradients-based methods, such as saliency maps [23] and Grad-CAM [22], have been widely used to explain model decisions in classification and prediction tasks. In the context of DR, sensitivity analysis helps reveal how perturbations in original features affect the positions of points in the 2D embedding. Techniques such as Probing Projections [24] and DimReader [7] link input features to the 2D projections, helping users understand cluster structures and local relationships [13]. Other studies explore how feature weighting affects projection layouts [2, 9, 11, 21]. Building on these ideas, our work constructs dynamic sensitivity fields rather than relying on pointwise or axis-based explanations, enabling both global and local interpretation through spatially coherent flow visualization.

Flow-based and Vector Field Visualization Flow visualization techniques, such as Line Integral Convolution (LIC) [3] and particle advection [25], have traditionally been used to represent physical vector fields like wind or fluid motion. Dynamic flow visualizations [10, 25], such as the "Wind Map" [8], have demonstrated the ability of animated motion to intuitively convey complex spatial patterns. Inspired by these techniques, we model feature sensitivity fields in DR projections as dynamic wind fields, allowing users to perceive feature sensitivity through continuous spatial flow.

3 Feature Wind Map Visualization

Background We measure each feature's importance by how sensitive the model's output is to small changes in that feature, i.e., by its gradient [20]. For a model f, the sensitivity of input x_i is defined as:

$$R_i = \left\| \frac{\partial}{\partial x_i} f(\mathbf{x}) \right\|. \tag{1}$$

When f is a DR mapping $\mathbb{R}^d \to \mathbb{R}^2$, each high-dimensional point $\mathbf{p}_i$ is projected as $\mathbf{v_i} = (v_{i,x}, v_{i,y})$, and for feature k we form the gradient as:

$$\mathbf{g}_{i,k} = \left[\frac{\partial f_x(p_i)}{\partial p_{i,k}}, \frac{\partial f_y(p_i)}{\partial p_{i,k}} \right]. \tag{2}$$

Stacking all $\mathbf{g}_{i,k}$ into a $2n \times d$ Jacobian matrix J, where n is the number of points, then captures how each 2D coordinate responds to perturbations in each of the input features:

$$\mathbf{J} = \begin{bmatrix} \mathbf{g}_{1,1}^T & \cdots & \mathbf{g}_{1,d}^T \\ \vdots & \ddots & \vdots \\ \mathbf{g}_{n,1}^T & \cdots & \mathbf{g}_{n,d}^T \end{bmatrix}. \tag{3}$$

In this paper, we use automatic differentiation(autodiff) to calculate $\mathbf{g}_{i,k}$. Autodiff augments programs to track a calculate the gradients as it the program executes. We employ the built-in Autodiff capabilities of the Torch library [5], which only requires that the DR program be compatible with torch tensors. It treats the projection as a generic differentiable computation graph and thus yields feature sensitivities whether the model is PCA, t-SNE, or a neural network. This plug-and-play gradient extraction removes the need for model-specific formulas, making Feature Wind Map inherently model-agnostic.

Vector-Field Formulation To construct a smooth visual encoding of $\mathbf{J}$, we first formulate a vector field over the DR projection space. Given a resolution r, we define a uniform grid with $r \times r$ cells over the region bounded by projected points $\{\mathbf{v}_i\}$.

Feature Wind Map begins by ranking all features by their average gradient magnitude. By default, Feature Wind Map only retains the top n features to focus attention on the attributes that significantly drive the embedding and to avoid diluting the visualization with minimum-impact gradients that add noise but little insight. And the fact that n is user-defined makes the system scalable from a multi-feature overview to a single-feature inspection.

For each feature k, we begin with its per-point gradient vectors for every point $\mathbf{v_i}$ (see Fig. 2A). We then interpolate the horizontal and vertical components of these vectors onto the grid vertices to obtain the vector field for feature k: $(\mathbf{U}_k, \mathbf{V}_k)$ (see Fig. 2B). Repeating this for all features produces a distinct vector field for each feature (see Fig. 2C). We combine them by summing all vectors in each grid vertex, which yields one global vector field $(\mathbf{U}, \mathbf{V})$ (vectors in Fig. 2D), where $\mathbf{U} = \sum_k \mathbf{U}_k$ and $\mathbf{V} = \sum_k \mathbf{V}_k$. To prevent Feature Wind Map from extrapolating flow into empty regions of far from the data, we build a KD-tree on each grid vertex c by computing its distance d_c to the nearest data sample. Cells with $d_c > \tau$ (a small threshold) are marked empty, and any particle seeded in or entering such cells is assigned zero velocity.

To determine the locally dominant feature, for each k, we compute it's magnitude field

$$\mathbf{M}_k = \sqrt{\mathbf{U}_k^2 + \mathbf{V}_k^2}, \tag{4}$$

then apply a Gaussian filter to obtain the smoothed fields $\widetilde{\mathbf{M}}_k$. At each grid point, the dominant feature index is

$$k^* = argmax_k \mathbf{M}_k. \tag{5}$$

Each grid vertex becomes a seed tagged with its dominant feature. Every pixel then inherits the label of the seed closest to it in Euclidean distance, yielding a Voronoi diagram that divides the projection into contiguous regions, each colored by the feature with the greatest local gradient magnitude. An example of feature dominance map is shown in Fig. 2D.

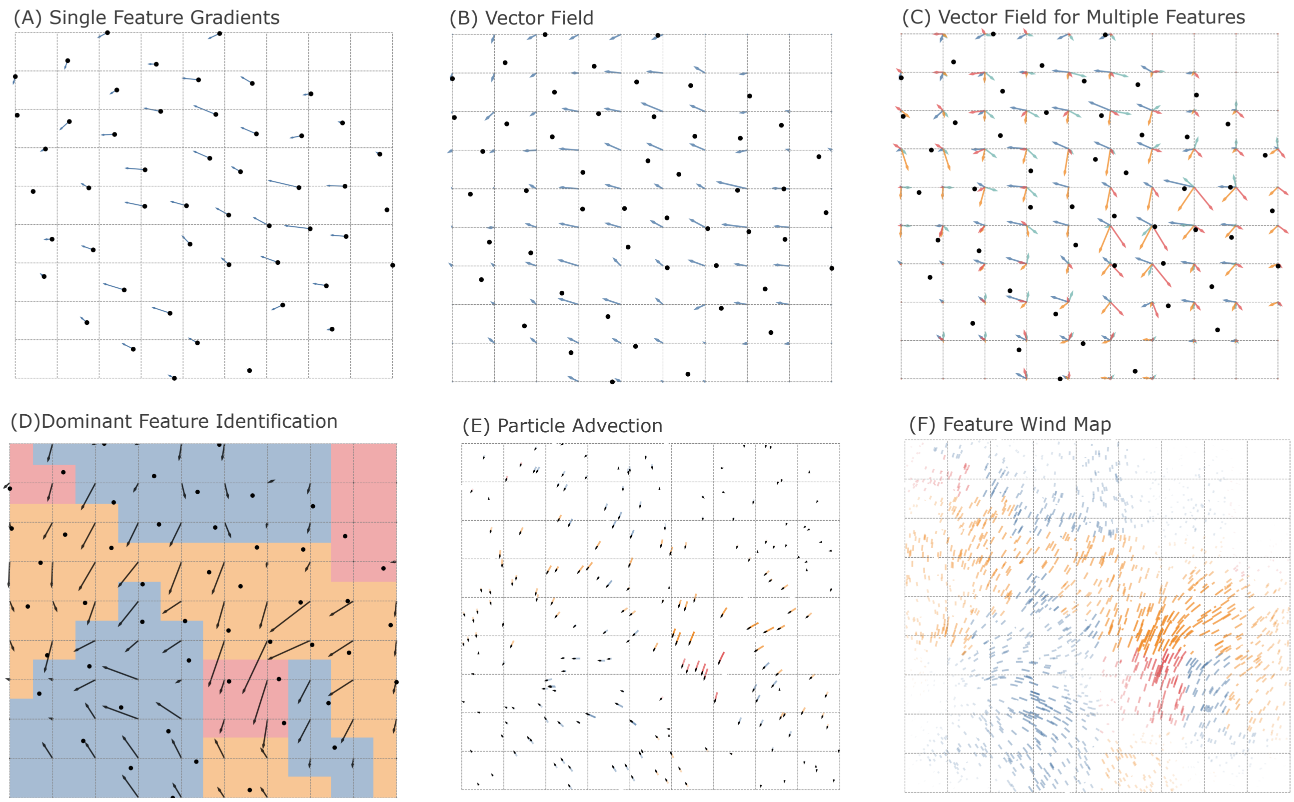

Figure 2: (A) point-wise gradients for one feature; (B) their interpolated vector field; (C) per-feature vector fields distinguished by color; (D) the aggregated field and a Voronoi overlay marking the locally dominant feature; (E) a single particle-advection frame whose black arrow shows flow direction; and (F) a snapshot of the final wind map rendered with 2000 particles.

Visualization Design We utilize particle advection to design the visualization of Feature Wind Map. We first randomly seed λ particles across the region bounded by $\{\mathbf{v}_i\}$. Each particle stores a short history of its previous positions (a "tail"). At each animation frame, for every particle at position **pp**, we determine the velocity **s** of the particle by interpolating from it's position in the vector field $(\mathbf{U}, \mathbf{V})$. Then the particle is advanced by $\Delta\mathbf{pp} = \alpha\mathbf{s}$, where α is a small constant that ensures smooth motion across frames. To depict flow, we connect each particle's previous and current positions with a line segment. We color each segment according to the particle's dominant feature at its current location, as shown in Fig. 2E (the black arrow indicates the direction of particle flow). To maintain a coherent flow visualization, any particle that leaves the bounding region or exceeds its maximum lifetime immediately re-initializes to a new random location. In addition, at every animation frame we randomly respawn 5% of the particles to prevent dead zones and keep the motion uniformly dense. The result (see Fig. 2F) is a high-density wind map, rather than static arrows, where the animated particle trails alone convey both direction and strength of the feature winds.

4 Case Study

To demonstrate the capabilities of Feature Wind Map, we evaluated it on both synthetic and real-world datasets. A video demonstration of Feature Wind Map is provided in supplementary material.

Synthetic Dataset This synthetic dataset comprises 100 points arranged along two rings in 3D. One ring, shown with circular markers, is parallel to the XY-plane, while the other, shown with square markers, is parallel to the XZ-plane (Fig. 3A). Fig. 3B presents a 2D t-SNE embedding of these points, overlaid with gradient vectors that indicate each point's sensitivity to the underlying features. This figure reveals that the ring parallel to the XY plane(left) is affected only by features X and Y, whereas the ring parallel to the XZ plane(right) is affected only by features X and Z. Also, from this figure, we see how sensitivities of different features change along the ring. However, in this figure each point is annotated with three overlapping gradients, creating visual clutter and making interpretation difficult. To remedy this, we apply Feature Wind Map to produce the wind map (Fig. 3C). By aggregating all local gradients into a continuous flow field, the wind map clearly highlights the global pattern and dominant feature in each region, an insight obscured by the cluttered gradients plot of Fig. 3B. Animated wind trails make the direction of feature sensitivity immediately readable, something overlapping static gradients cannot achieve.

Breast Cancer Dataset The Wisconsin Breast Cancer dataset [28] comprises 569 samples derived from digitized images of fine-needle aspirates of breast masses. For each sample, 30 real-valued features are computed, capturing cell-nuclei morphology such as mean radius, texture, smoothness, perimeter, area, compactness, concavity, and more. Each example is labeled as either benign (marked as square) or malignant (marked as circle) in Fig. 1. Because its features have clear physical interpretations and the dataset is sufficiently complex, it has become a standard benchmark for evaluating not just predictive accuracy but also methods for DR explainability. We apply Feature Wind Map to the t-SNE projection of the breast cancer dataset to demonstrate it's ability to provide insight into the projection. In this example, we select the top 5 features to aggregate in the feature wind map.

Global Structure: The wind map's overall flow trails highlight the separation between benign and malignant clusters in the 2D em-

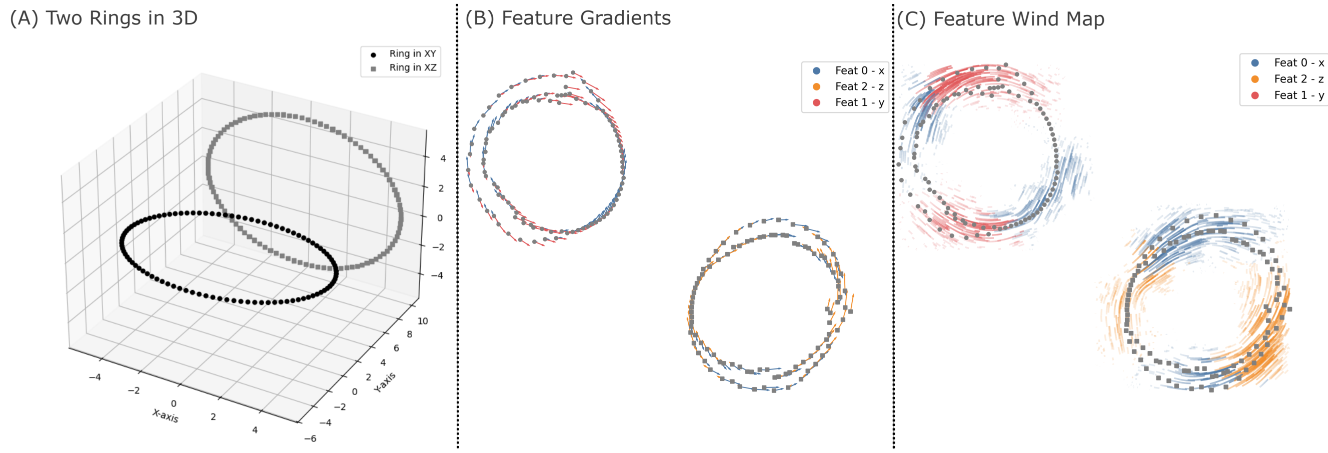

Figure 3: (A) shows original 3D dataset; (B) visualizes each feature's gradients; (C) is the applied Feature Wind Map.

bedding. In Fig. 1C, a rough decision boundary is identified where winds flow across the two classes. Along this boundary, the *worst texture* and *worst smoothness* features dominate the flows, indicating that small perturbations in these two features exert significant shifts on t-SNE's decision. Furthermore, the opacity and density of winds suggest that malignant clusters have lower sensitivity than the benign cluster. Strong winds in the benign region tells us that small changes in these features cause large movements in the projection. It flags a high degree of internal sensitivity within the benign samples, which t-SNE alone does not make obvious.

Regional Patterns: Beyond simply showing where features have high sensitivity, Feature Wind Map also organizes the projection into spatially coherent zones. In Fig. 1C, Feature Wind Map colors each wind trail by its locally dominant feature, which allows insights into blocks of the embedding sharing the same color. These contiguous "color-blocks" correspond to regions where a single feature exerts the greatest pull on the embedding, partitioning the DR space into interpretable subdomains. It helps users visually segment the map into areas governed by different attributes.

Single Feature vs. Multiple Feature: Fig. 1B shows how *worst texture*, which has the highest sensitivity across the embedding, would affect the DR projection. It allows users to directly explore the sensitivity pattern of a single feature along the projection. Notice that even though *worst texture* has overall highest sensitivity, it may not be the dominant feature in every region. For example, *worst texture* possesses large sensitivity in the area highlighted in Fig. 1B-1, but if we look at sensitivity pattern from multiple features (Fig. 1C), *mean texture* actually dominates this area. On one hand, single-feature maps provide feature-centric inspection, which can answer question such as "What happens if I tweak just this feature?". On the other hand, the multi-feature map delivers a holistic overview of global feature patterns, ideal for question like "What feature has the highest sensitivity in this region, and where do boundaries shift?"

5 Discussion and Future Work

Direction of Sensitivity: DimReader [7] visualizes sensitivity by drawing a local axis orthogonal to a selected feature's gradient (see Fig. 4), while Feature Wind Map visualizes wind trails following the sensitivity direction. Visualization along the gradient follows the direction in which a perturbation of that feature would push a point in the embedding, so the flow orientation directly answers the "what-if" question: if this feature has a tiny increase, where will the point go? Plotting an axis orthogonal to the gradient instead shows an isoline along which the feature value stays constant, which inverts the mental model because motion is now implicit and must be inferred from the perpendicular axis. By moving along the actual gradient, Feature Wind Map displays sign, strength, and inter-feature competition in one coherent flow, while orthogonal axes suit only single-feature probes and elide the precise displacement direction, requiring one more cognitive step.

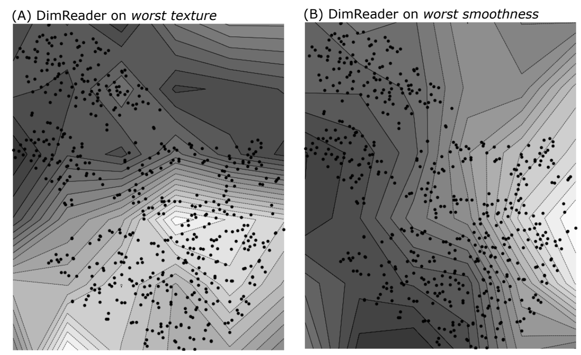

Figure 4: DimReader recovers axes that reveals sensitivity patterns of high-dimensional features. A perturbation of the related feature will push the point from lighter area to darker area.

Value of Adding Motion to Visualization: Animation turns sensitivity from a static picture into a visual experience that our perceptual system readily decodes [1]. Moving particles instantly convey direction, speed, and convergence, so viewers grasp flow patterns at a glance instead of deciphering a thicket of arrows. Because trails fade and renew, the display streams richer information through time than any single frozen frame. Motion also sharpens boundaries: as particles cross zones with different colors, their hue flips on cue, clearly marking where feature dominance changes.

Future Work: Potential future work could add interactive filtering and steering of particles for scalable exploration of very larger datasets with extensive numbers of features. Also, extending the current visualization to 3D spaces seems promising as well.

6 Conclusion

In this paper, we present Feature Wind Map, a flow-based visualization method to illustrate DR sensitivities, generated from feature gradients. Feature Wind Map aggregates feature gradients into a unified vector field and leverages particle advection to demonstrate the flow of vectors across the projection. The resulting flow illustrates the sensitivity of the DR to changes in the underlying features. We demonstrated Feature Wind Map on both synthetic and real-world datasets through two case studies, highlighting it's ability to recover the driving features in the DR mapping and uncover decision ridges and class-specific sensitivities.

References

[1] L. Bartram. Perceptual and interpretative properties of motion for information visualization. In *Proceedings of the 1997 Workshop on New Paradigms in Information Visualization and Manipulation*, NPIV '97, p. 3–7. Association for Computing Machinery, New York, NY, USA, 1997. doi: 10.1145/275519.275520 2, 4

[2] Y. Bian, J. Wenskovitch, and C. North. Deepva: Bridging cognition and computation through semantic interaction and deep learning. In *2019 IEEE Workshop on Machine Learning from User Interaction for Visualization and Analytics (MLUI)*, pp. 1–10. IEEE, 2019. 2

[3] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 263–270, 1993. 2

[4] S. Cheng and K. Mueller. The data context map: Fusing data and attributes into a unified display. *IEEE transactions on visualization and computer graphics*, 22(1):121–130, 2015. 1

[5] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. 2

[6] R. R. da Silva, P. E. Rauber, R. M. Martins, R. Minghim, and A. Telea. Attribute-based visual explanation of multidimensional projections. In *EuroVA15*. The Eurographics Association, 2015. 1

[7] R. Faust, D. Glickenstein, and C. Scheidegger. Dimreader: Axis lines that explain non-linear projections. *IEEE transactions on visualization and computer graphics*, 25(1):481–490, 2018. 1, 2, 4

[8] M. W. Fernanda Bertini Viégas. Wind map, 2012. 2

[9] H. Kim, J. Choo, H. Park, and A. Endert. Interaxis: Steering scatterplot axes via observation-level interaction. *IEEE transactions on visualization and computer graphics*, 22(1):131–140, 2015. 2

[10] R. S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf. The state of the art in flow visualization: Dense and texture-based techniques. In *Computer graphics forum*, vol. 23, pp. 203–221. Wiley Online Library, 2004. 2

[11] J. Lin, R. Faust, and C. North. Imagesi: Semantic interaction for deep learning image projections. In *2024 IEEE Visualization and Visual Analytics (VIS)*, pp. 91–95. IEEE, 2024. 2

[12] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE transactions on visualization and computer graphics*, 23(3):1249–1268, 2016. 1, 2

[13] W. Liu, C. North, and R. Faust. Visualizing spatial semantics of dimensionally reduced text embeddings. *arXiv preprint arXiv:2409.03949*, 2024. 2

[14] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 1

[15] B. Montambault, G. Appleby, J. Rogers, C. D. Brumar, M. Li, and R. Chang. Dimbridge: Interactive explanation of visual patterns in dimensionality reductions with predicate logic. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 1

[16] O. Ovcharenko, R. Sevastjanova, and V. Boeva. Feature clock: High-dimensional effects in two-dimensional plots. In *2024 IEEE Visualization and Visual Analytics (VIS)*, pp. 151–155. IEEE, 2024. 1, 2

[17] K. Pearson. Principal components analysis. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 6(2):559, 1901. 1

[18] D. H. Pilar and C. Ware. Representing flow patterns by using streamlines with glyphs. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1331–1341, 2013. doi: 10.1109/TVCG.2013.10 2

[19] M. Rautenhaus, M. Böttinger, S. Siemen, R. Hoffman, R. M. Kirby, M. Mirzargar, N. Röber, and R. Westermann. Visualization in meteorology—a survey of techniques and tools for data analysis tasks. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3268–3296, 2018. doi: 10.1109/TVCG.2017.2779501 2

[20] W. Samek. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017. 2

[21] J. Z. Self, M. Dowling, J. Wenskovitch, I. Crandell, M. Wang, L. House, S. Leman, and C. North. Observation-level and parametric interaction for high-dimensional data analysis. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):1–36, 2018. 2

[22] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, 128:336–359, 2020. 2

[23] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 2

[24] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE transactions on visualization and computer graphics*, 22(1):629–638, 2015. 1, 2

[25] A. Sundquist. Dynamic line integral convolution for visualizing streamline evolution. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):273–282, 2003. 2

[26] B. Tversky, J. B. Morrison, and M. Betrancourt. Animation: can it facilitate? *International journal of human-computer studies*, 57(4):247–262, 2002. 2

[27] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 1

[28] W. Wolberg, O. Mangasarian, N. Street, and W. Street. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository, 1993. DOI: https://doi.org/10.24432/C5DW2B. 3