

---

Article

Not peer-reviewed version

---

# A General Framework for Visualizing Machine Learning Models

---

Ziqian Bi , Raymond Gao , [Shiaofen Fang](#) \*

Posted Date: 14 February 2024

doi: [10.20944/preprints202402.0798.v1](https://doi.org/10.20944/preprints202402.0798.v1)

Keywords: machine learning; multi-dimensional data visualization; classification; morphing; scattered data interpolation



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

## Article

# A General Framework for Visualizing Machine Learning Models

Ziqian Bi <sup>1</sup>, Raymond Gao <sup>2</sup> and Shiaofen Fang <sup>1,\*</sup>

<sup>1</sup> Luddy School of Informatics, Computing, and Engineering, Indiana University Indianapolis

<sup>2</sup> Acton-Boxborough Regional High School, Acton, Massachusetts

\* Correspondence: shfang@iu.edu

**Abstract:** As Artificial Intelligence (AI) technology, particularly Machine Learning algorithms, becomes increasingly ubiquitous, our abilities to understand and interpret AI and machine learning algorithms are becoming increasingly desirable. Visualization has been used as a common tool to view and understand complex machine learning processes. In this paper, we will focus on developing a general framework for the visualization of machine learning models as a scalar valued multi-dimensional function to help users understand how the models behave over different viewing spaces. We first give a formal definition of the visualization problem. Interpolation-based morphing and subspace sampling techniques are applied to generate various renderings through projections and cross-sections of the model space as 3D surfaces or heatmap images. This method will be applied to two real-world datasets and applications: the diagnosis of Alzheimer's Disease (AD) using a human brain networks dataset and a real-world benchmark dataset for predicting home credit default risks. The visualizations show that different machine learning algorithms can behave quite differently under different conditions.

**Keywords:** machine learning; multi-dimensional data visualization; classification; morphing; scattered data interpolation

---

## 1. Introduction

Machine learning algorithms act mostly as a black box, i.e., the users have very little information about how and why the algorithms work or fail. The underlying machine learning models are also designed primarily for the convenience of learning from data, but they are not easy for the users to understand or interact with. Explainable Artificial Intelligence, particularly explainable machine learning algorithms, is a critical area to ensure safety and trust in the use of AI technologies in human society [1]. One of the most powerful tools in developing explainable machine learning algorithms is visualization [2]. Being able to view the progression of a decision-making process in a machine learning algorithm is often a desirable feature for many critical AI applications, particularly when deep neural networks are used [3]. While visualizing the process of an algorithm can provide useful insight about the decision-making steps of the machine learning model, it would also be beneficial if visualization can be used to show the overall shape pattern of the machine learning model itself in some space that the users can understand. This type of model visualization has not been sufficiently studied, primarily because it is very challenging to visualize a high-dimensional function (as is the case with most machine learning models) in a limited screen space.

Although high-dimensional data visualization techniques [4] can also be applied to a set of sample points computed by the model in the high-dimensional space, the fact the model represents a continuous function with intrinsic shape information cannot be captured using traditional information visualization techniques for discrete data set. Some types of rendering methods are necessary to represent the continuous shape patterns.

We focus on machine learning models that can be defined as a scalar valued function in a high-dimensional feature space, i.e., supervised single valued model trained using a training dataset. The training samples can also play an important role in the visualization process. To this end, we can use volume visualization as an analog when considering this visualization problem [5]. A typical volume data such as a CT or MRI volume is a single valued function defined over a 3D domain. If we extend the 3D domain to an N-dimensional feature space, it defines a machine learning model where the function value is the learning label such as the classification probability or value of a predictive regression model. The rendering of such a model is, however, more challenging for several reasons. First, the concepts of depth cue and visual perception do not exist in high-dimensional space. Therefore, traditional rendering operations such as blending and shading do not apply. Secondly, sampling in a higher dimensional orthogonal subspace (for each pixel) to the viewing space does not have a simple order. Thus, cross-sections and projections will need to be carefully re-defined to generate meaningful visual representations. Third, when the dimensionality of the feature space is high, a 2D screen space is a very narrow and limited viewing window. Thus, the selection of and interaction with the viewing spaces are important for the understanding and interpretation of the model.

This paper proposes a general framework for the visualization of a large class of machine learning models (supervised single valued model trained using a training dataset). Our approach provides multiple perspectives (viewing spaces) and viewing options to allow the user to gain an insight into the shape pattern of a machine learning model. The ability to visualize a machine learning model is a critical step in facilitating interactive machine learning and human-in-the-loop AI [6]. Model visualization provides several ways for the users to interact with the machine learning algorithm to better understand and improve the learning process [7]. It has several important potential benefits:

- 1) Interpretability. Visualization can help the user to interpret and explain how decisions are made using different machine learning models. It is also useful to view different behaviors of different types of models so that users can make more informed decisions with realistic expectations.
- 2) Interaction. Human knowledge and intuition can significantly improve the performance of machine learning algorithms. Visualization is essential in interactive systems, where human feedback and interferences can help improve the machine learning process in the form of feature selection, parameter setting, or rule revision.
- 3) Training Data. The way training data is used in a machine learning algorithm can significantly impact the performance of the algorithm. Model visualization can provide a visual aid for interactive selection of training data in certain situations such as data reduction and bias mitigation. For example, properly selected training data can significantly reduce the size of the training set required to reduce cost and training time [8]. Carefully managed distributions of the training sets can also help mitigate potential biases of machine learning models [9]. These applications will require the visualization of the ML models in a space that the user can easily understand and take actions, for example, by adding certain types of samples defined by the variables of the viewing space.

In the following, we will first, in Section 2, discuss previous work related to visualization in AI and machine learning, including visual analytics and related morphing and rendering techniques. Section 3 provides an outline and a formal definition of our model visualization framework. Section 4 discusses details of several specific machine learning model visualization techniques and methods. We will use two real-world datasets to show the results and comparisons of the various visualization methods for different models in Section 5. Conclusions and future work are given in Section 6.

## 2. Related Work

Applying visualization and visual analytics principles in interactive or human-in-the-loop machine learning has become an active research area in recent years [2]. Most of the existing studies focus on using visualization for better understanding of the machine learning processes [3]. There

has also been some recent work on using visual analytics for improving the performance of machine learning algorithms through better feature selection or parameter setting [11,12].

Techniques in visual analytics have been widely used to accomplish analysis tasks such as classification and regression [13,14]. These techniques include mostly direct visualization methods without explicit machine learning, though machine learning algorithms can sometimes also play a supplementary role in a visual analytics system [11]. In this paper, we primarily focus on visualization methods for machine learning algorithms and models [15].

Previous works on using visualization to help understand the machine learning processes are usually designed for specific types of algorithms, such as support vector machines, neural networks, and deep learning neural networks. Neural Networks received the most attention due to the complexity of their internal components. Multi-dimensional visualization techniques such as scatterplot matrix have been used to depict the relationships between different components of the neural networks [16,17]. Typically, a learned component is represented as a higher dimensional point. The 2D projections of these points in either principal component analysis (PCA) spaces or a multi-dimensional scaling (MDS) space can better reveal the relationships of these components that are not easily understood, such as clusters and outliers. Several methods apply graph visualization techniques to visualize the topological structures of the neural networks [18–20]. Visual attributes of the graph can be used to represent various properties of the neural network models and processes.

Several recent studies addressed the challenges of visualizing deep neural networks. In [21], Liu et al. developed a visual analytics system, CNNVis, that helps machine learning experts understand deep convolutional neural networks by clustering the layers and neurons. Edge bundling was also used to reduce visual clutter. Techniques have also been developed to visualize the response of a deep neural network to a specific input in a real-time dynamic fashion [22,23]. Observing the live activations that change in response to user input helps build valuable intuitions about how convnets work. There are several literatures that discuss visualization's roles in Support Vector Machines. In [24], visualization methods were used to provide access to the distance measure of each data point to the optimal hyperplane as well as the distribution of distance values in the feature space. In [25], multi-dimensional scaling technique was used to project high-dimensional data points and their clusters onto a two-dimensional map maintaining the topologies of the original clusters as much as possible to preserve their support vector models. In [26], interactive volume visualization was used to identify potential features for classification of brain network data. Finally, Visualization and visual analytics methods have also been used to analyze the performances of machine learning algorithms in different applications [27–29].

Compared to the visualization of machine learning processes, there have been relatively few known techniques for the visualization of general machine learning models. The Manifold system [30] provides a generic framework that does not rely on or access the internal logic of the model and solely observes the input and output. It applies scatter plot matrix visualization to observe input and output samples to evaluate model performance and behavior. In [8], scatter plots were also used to visualize machine learning models to help select the optimal set of training samples. In this paper we propose a new approach for the visualization of any general machine learning model as a feature space function. Our approach applies a surface morphing technique that is often used in shape deformation [31,32]. A few scattered data interpolation methods described in [33] are used in these morphing techniques in our model visualization.

Using subspace concepts to visualize high-dimensional datasets has been explored in information visualization. 2D linear projections from unique linear subspaces are used to visualize high-dimensional data in [35]. Singular value decomposition is applied to the high dimensional data to detect 1D subspaces for effective search and exploration of generative models [36]. In [37], topological and geometric techniques are used to approximate the high dimensional data by Morse-Smale complex on the cloud of point samples through parametric space segmentation. A simplified geometric representation of the Morse-Smale complex is then visualized by 2D embedding. These techniques are designed for the projection of discrete point data rather than a continuous model or manifold where discrete points cannot capture the true and continuous shape information. Geometric

and topological approximation is also problematic as details of the model, even if not smooth (e.g., rough boundaries), are important information for interactive ML (e.g., the need to add points in rough boundary areas). A good analog is the ray casting volume rendering algorithm [5] where linear rays are sampled (in different ways) and projected to each pixel of the viewing plane. But in high-dimensional situation there is no perspective projection as 3D human perception does exist in high-dimensional space.

### 3. Outline of the Framework

Our goal is to develop a multi-view visualization technique to generate multiple perspectives to help user gain a comprehensive visual impression of a machine learning model's overall behavior or tendency. In this section we will first outline a formal definition of the visualization framework.

A machine learning model can be defined as a function:

$$F(x_1, x_2, \dots, x_n) : R^n \rightarrow R$$

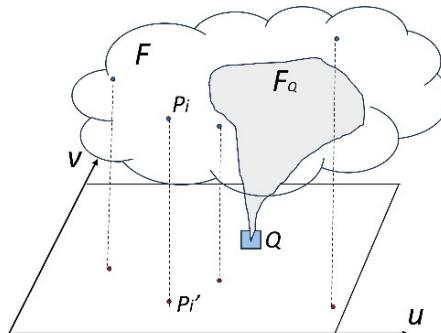
where  $R^n$  is the n-dimensional feature space, and the output is the result of the machine learning algorithm which can be either a classification probability or a predicted regression function value. We also assume that the machine learning model is trained using a training set:

$$P = \{P_1, P_2, \dots, P_k\} \subset R^n \text{ and } \{F(P_i) : i = 1 \dots n\} \subset R$$

The visualization problem is hereby defined as an image on a 2D viewing space  $(u, v)$ , representing some information about the function  $F$ , projected onto this 2D viewing space. The viewing space is generally a 2D subspace of the feature space. The meaning of projection here has two components:

1. Subspace determination: For each pixel in the viewing space, find the subspace in the feature space that are orthogonal to this pixel point. This is the information that needs to be represented by this pixel through projection.
2. Subspace sampling: Within the orthogonal subspace, determine what values should be used to render this pixel, either as a colour or as a height value for generating a surface. This is the process of sampling or information filtering and integration for visual presentation.

Essentially, we consider a machine learning model an n-dimensional function of scalar values. We also consider the training set an integral part of the function. The visualization problem becomes a problem of projecting the n-dimensional function onto selected 2D viewing spaces. The high dimensionality makes this projection less defined and under constrained. Therefore, many of the details of the algorithms are about how to find the proper information to project and draw. Apparently multiple-viewing spaces and potentially animation-assisted techniques will be necessary to gain sufficient visual representation for a good understanding of the model's behavior and patterns. These viewing dynamics can come from different viewing spaces selected by the users, as well as animated or interactively selected information filtering parameters. Figure 1 shows a summary of this framework for a viewing space  $(u, v)$  and a general machine learning model  $F$ .



**Figure 1.** A summary illustration of the visualization approach,  $F$  is the machine learning model;  $(u, v)$  is the viewing space;  $P_i$  are the training set samples. Their projections on  $(u, v)$  space are  $P_i'$ ;  $F_Q$  is

the subspace orthogonal to the pixel Q. This subspace will be projected onto Q after information filtering and sampling.

#### 4. Visualization Approach

There are three main components in our visualization approach: viewing space selection, subspace determination, and rendering by subspace sampling.

##### 4.1. Viewing Space Selection

Viewing space,  $(u, v)$ , is a 2D subspace of the feature space which the machine learning function will be projected onto. The user will select two variables to represent the 2D axes of the viewing space. In most situations, multiple viewing spaces will need to be selected to provide multiple views and perspectives. Two separate criteria can be used to choose the viewing space variables.

Interpretable variables. These are usually the features that the users are familiar with, thus, can be used to better understand the behavior of the machine learning model. Typical examples include demographics information, or features that can be easily obtained such as test scores and some simple measurements, but not something abstract or calculated by a complex process.

Representative variables. These are the features or variables (can be combinations of features) that can capture the most amount of information or variations of the machine learning model. A good example would be the Principal Components obtained from a Principal Component Analysis (PCA) process. We will assume in this paper that these variables can only be linear combination of original features since it becomes difficult to find the orthogonal subspace for nonlinear combinations.

A viewing space defined by interpretable variables is good for users to interpret or explain the behavior of the model, but they are sometimes not the best projection “angle”, i.e., the amount of information the viewing space can capture may not be sufficient to truly understand changes or tendencies in the model representation. This type of viewing space is also important if new training samples are required as a result of the visualization, as it is easier for users to collect new training samples using common and easy to measure features.

On the other hand, a viewing space defined by representative variables can be a better “angle” to maximize the amount of information projected. However, the variables may not be intuitive for the purpose of explaining or interpreting the results. Therefore, these two types of viewing spaces complement each other and should be used together.

##### 4.2. Viewing

For each point or pixel on a viewing space, the first task is to determine the subspace in the feature space that is orthogonal to this point. This is the subspace that should be projected onto this pixel location when the entire model is projected to the viewing space. As a general form of viewing space representation, let:

$$\begin{cases} u = a_1x_1 + a_2x_2 + \dots + a_nx_n \\ v = b_1x_1 + b_2x_2 + \dots + b_nx_n \end{cases} \quad (1)$$

be the viewing space axes. In this paper we will focus only on linear projections to include dimension reduction methods such as traditional Principal Component Analysis (PCA). The primary reason for only focusing on linear projections is interpretability. The visualization results need to be easy to understand by the end users. The most common viewing space will be a 2D space of two original features with clear meanings. Linear dimension reduction space sometimes can be useful as well as an overview. But nonlinear projection spaces (e.g., some multi-dimensional scaling sapces) are too difficult for the end users to interpret and make sense (though they may have values in some other applications).

In equation (1), when all  $a_i$  are zeros except one and all  $b_i$  are zeros except one, then  $(u, v)$  are simply two of the original features, which may be two interpretable features. A more general case would be the PCA space, i.e.,  $u$  and  $v$  can be the two most important (with largest eigenvalues) principal components of the dataset. This ensures that the projection retains the maximum amount of information about the machine learning model.

The orthogonal subspace can be generated by solving the linear equation system (1) for  $(u, v)$  in two steps:

- Identify two dominant variables,  $x_l$  and  $x_m$  in the equation system, where

$$a_l = \max_i a_i \text{ and } b_m = \max_{i \neq l} b_i$$

- Solve the equation system with respect to the two variables,  $x_l$  and  $x_m$ :

$$\begin{cases} x_l = c_0 + \sum_{i \neq l, i \neq m} c_i x_i \\ x_m = d_0 + \sum_{i \neq l, i \neq m} d_i x_i \end{cases} \quad (2)$$

where  $c_i$  and  $d_i$  are constant coefficients. The subspace is then defined by the set of all points in the feature space that satisfy the equation (2) and can be projected onto the given pixel location  $(u, v)$ . A special case is when  $u = x_l$  and  $v = x_m$ . Then the equation becomes  $x_l = u$  and  $x_m = v$ .

In summary, for each pixel location in a viewing space, if the viewing space axes are part of the original features (e.g., interpretable variables), the subspace is all points carrying the pixel location coordinates for the two features that define the viewing space. Otherwise, the subspace is the set of all points with coordinates satisfying equation (2).

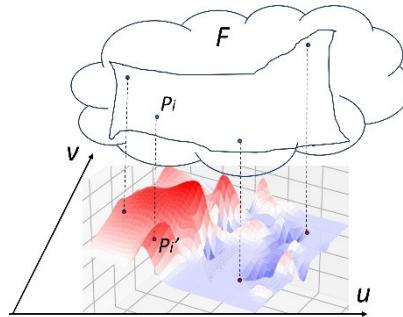
#### 4.3. Rendering by Subspace Sampling

Subspace sampling refers to the filtering and integration of information in the subspace for drawing all the pixels in the viewing space. There are infinite number of ways to sample and select information in a subspace. In the following, we will propose several different techniques that can provide multiple visual perspectives of a machine model.

For each viewing space  $(u, v)$ , our approach will generate various surfaces or heatmap images defined over  $(u, v)$  to represent various types of projections or cross-sections of the machine learning model. In the following we will discuss two different ways of constructing this surface or image: morphing by interpolation, and subspace projection.

##### 4.3.1. Morphing by Interpolation

This method considers the fact that a machine learning model is trained using a training set. Therefore, points in the training set can be considered key points that drive the shape of the machine learning function. Key points based shape morphing technique can then be used to “deform” the function  $F$  to fit into the viewing space. In this case, of course, the morphing process is not between spaces of the same dimensions. A morphological deformation from a high-dimensional space to a 2D viewing space does not maintain all the shape information of the manifold. But it can be viewed as a cross-section by a 2D shape (i.e., a curved surface) that passes through all the key points, and thus captures the most important shape variations, as illustrated in Figure 2:



**Figure 2.** Morphing illustration. The surface projected to the viewing space is constructed in the features space by interpolating the training samples based on their positions on the viewing space. The interpolation for each pixel is performed within that pixel's subspace.

This morphing and interpolation algorithm is formulated as follows: Let  $P = \{P_1, P_2, \dots, P_k\} \subset \mathbb{R}^n$  be the training samples. Their projections onto  $(u, v)$  are  $P' = \{P'_1, P'_2, \dots, P'_k\} \subset \mathbb{R}^2$ . For each pixel location  $Q$ , its subspace in the feature space is defined by equation (2). An interpolation function is then constructed to find the feature values for the free variables in equation (2):

$$x_i = f(Q, P') \quad (i \neq l, i \neq m)$$

where function  $f$  can be any scattered data interpolation function [33]. Combined with  $x_l$  and  $x_m$ , as shown in equation (2), these features values form a complete feature vector  $V$  for each pixel. The value of  $F(V)$  is then assigned to the pixel as the z-coordinate of the surface. In our implementation, we tested two interpolation functions: linear interpolation and affine Shepard interpolation.

In linear interpolation, the training samples on the 2D space,  $P' = \{P'_1, P'_2, \dots, P'_k\}$ , are triangulated by Delaunay triangulation first. Within each triangle a linear interpolation using barycentric coordinates is applied to generate weights that are used to interpolate the feature space variables.

In affine Shepard interpolation, we modified the classic Shepard interpolation for scattered data by adding a local affine function, or a plane, at each key point to avoid discontinuities at the interpolated points:

$$f(Q) = \sum_1^n \frac{g_i(Q)}{d_i^r(Q, P'_i)} \Bigg/ \sum_1^n \frac{1}{d_i^r(Q, P'_i)}$$

where  $d_i(Q, P'_i) = \text{distance}(Q, P'_i)$ ,  $r$  is an adjustable parameter (default is 2), and  $g_i(Q)$  is a function of a plane that passes through  $P'_i$  and is parallel to a local triangle formed by the nearest 3 key points.

#### 4.3.2. Subspace Projection

Morphing method can capture the shape variations defined by the training samples, but it only provides one cross-section of a high-dimensional feature space. For a more comprehensive view, we can generate a large set of points in each subspace and visualize various subsets of these points to show the distribution of values in this subspace. This represents different ways to project information to the 2D viewing space.

For each pixel  $Q$  in  $(u, v)$ , its subspace in the feature space is defined by the free variables in equation (2), i.e.,

$$S_Q = \{x_i: i \neq l, i \neq m\}$$

This is an  $n-2$  dimensional space. When  $n$  is large, regularly sampling even a coarse grid would be prohibitive since every sample will need to be fed into the machine learning function. One alternative is to randomly sample a pre-determined number of points. This seems to work reasonably well in our experiments as the visualization results seem to be consistent with multiple times of sampling. Assuming  $N$  samples are taken in each subspace:

$$\{X_i: i = 1 \dots N\} \subset S_Q$$

and let the features vectors after adding the two non-free variables  $x_l$  and  $x_m$  be:

$$\{Y_i: i = 1 \dots N\} \subset \mathbb{R}^n$$

we can then select different subsets of  $\{F(Y_i)\}$  to show at the original pixel location.

One approach we take is to sort  $\{F(Y_i)\}$  values from high to low, and then select sequence of given percentile values to draw. This will give the users a clear understanding of the distribution of the machine learning results for this pixel value. The combined surface or heatmap image over  $(u, v)$  will show a global pattern and trends of the performance of the model. One special case is the 100-percentile surface. It represents the largest value in each subspace, similar to the Maximum Intensity Projection (MIP) technique in medical volume visualization.

Another way to access the information in each subspace is to generate a histogram of the values  $\{F(Y_i)\}$  for each pixel. This process creates a histogram volume over the entire viewing space  $(u, v)$ . Cross-sections of this histogram volume show the concentrations (number of samples) at different values (e.g., probabilities) across the viewing space. This may be better rendered using 2D heatmap images rather than surfaces.

A third possibility is to display an average value of the subspace for each pixel. Please note that this is different from the 50 percentile value.

Not all these types of projections will be useful for all models. Some may show interesting patterns in one but not the others. Please also note that the color mapping scheme we use in this paper is not consistent across different surfaces. Each image / surface uses its own color map based on its value range. This is because the value (e.g., Probability) range within each surface is very small. A global color scheme will create very little color contrast within each surface. So, we redefine the color map for each visualization image to maximize the color contrast within its own value range.

## 5. Experimental Results

### 5.1. Datasets

We applied our visualization framework on two real-world applications: the diagnosis of Alzheimer's Disease (AD) using a human brain networks dataset obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database ([adni.loni.usc.edu](http://adni.loni.usc.edu)), and a real-world benchmark dataset for predicting home credit default risks.

The primary goal of ADNI has been to test whether various modalities of brain images, other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early AD. The original data included both structural MRI and diffusion tensor images (DTI). A separate tractography technique was used to generate a connectome network for each subject to measure the connectivity of different regions of interest (ROIs) in a human brain [34]. The connectome network is modeled as an undirected graph with ROIs in the brain as graph nodes and DTI fiber density as edge weights. We calculate the degree of each node (ROI) as the sum of weights of all connected edges to this node. These degrees are used as the initial features for machine learning systems. We also added several additional common features for each subject: age, education level, BMI, and MMSE (Mini-Mental State Examination) score. There are 158 subjects in 3 categories: HC (Healthy Control, 58 subjects); MCI (Mild Cognitive Impaired, 71 subjects) and AD (Alzheimer's Disease, 29 subjects). Each subject's connectome network has 100 node degree features and 4 additional common features, totaling 104 features. The age range of these subjects is from 55 to 90.

The second dataset is a real-world benchmark dataset collected by Home Credit, the Home Credit Default Risk dataset (<https://www.kaggle.com/c/home-credit-default-risk/overview>). It includes a variety of statistical information from the clients, such as biometric information, credit history, etc. We built a model based on this dataset to predict the clients' repayment abilities, where the predicted result 1 represents that the client has payment difficulties and 0 represents all other cases. The dataset we use includes 10,000 samples, among which 5000 are positive (label 1) and the other 5000 are negative (label 0).

### 5.2. Machine Learning Models

A machine learning algorithm can be applied to a predictive model for each dataset. For the ADNI dataset, the 3-class (HC, MCI, AD) classification problem is defined as a regression model. We assign 0 to HC label, 0.5 to MCI label, and 1 to AD label. A value returned from a machine learning regression model can be used to classify a subject into one of the three classes based on the three class intervals: HC=[0,0.33], MCI=(0.33,0.67), and AD=[0.67,1]. A binary classification model is trained for the Home Credit dataset.

We applied three popular machine learning algorithms: Support Vector Machine (SVM) (or Support Vector Regression (SVR)), Extreme Gradient Boosting (XGBoost), and Deep Learning (DL). The overall prediction accuracies, F1 scores, and AUC scores for both datasets are given in Table 1. These algorithms represent different styled machine learning strategies. It can be interesting to see how these models look in various viewing spaces. The differences in accuracy and other performance metrics for the three models are not significant here as we did not do extensive parameter optimization for performance purposes.

**Table 1.** Three machine learning models' performance data.

	Accuracy		F1 Score		AUC score	
	ADNI	Credit	ADNI	Credit	ADNI	Credit
SVM	0.73	0.62	0.73	0.62	0.89	0.65
XGBoost	0.72	0.68	0.72	0.67	0.88	0.74
DL	0.70	0.60	0.70	0.64	0.88	0.64

### 5.3. Visualization

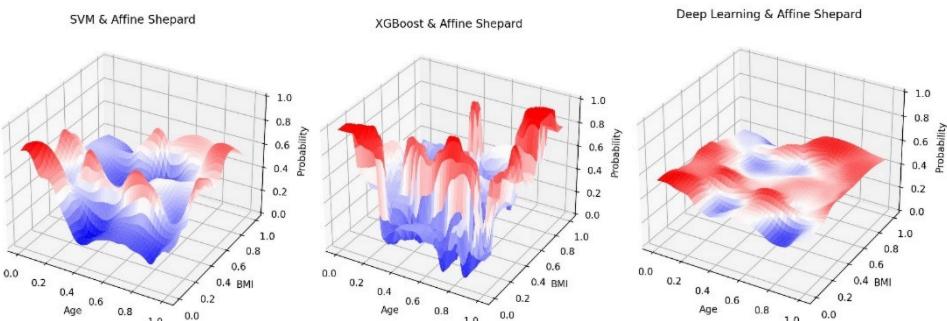
#### 5.3.1. Viewing Spaces

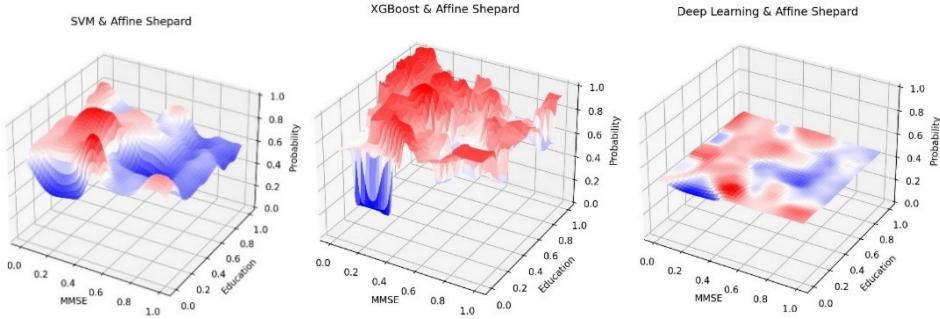
The first step in visualization is to determine the potential viewing spaces. For this ADNI dataset, the 4 common features, Age, Education Level, BMI and MMSE Score, are clearly interesting and interpretable variables that we can use to view the machine learning models. For the Home Credit dataset, most of the features are meaningful variables that the users can easily relate to. Thus, we can use the features that have larger weights in the machine learning algorithms such as income level, credit amount in a loan, home conditions (EMERGENCYSTATE\_MODEL), and price of the goods to buy using the loan. For an overview visualization with more representational power, we can use the top two principal components, PC0 and PC1, from a PCA transformation of the feature space. It is of course also possible to select any two features that the users are interested in. For example, in the ADNI dataset, since the features represent connectome network nodes, this type of viewing space can provide useful information about specific ROIs of the human brain that the neuroscientists may be able to interpret.

In the following, we will show various visualizations of the machine learning models (SVM, XGBoost and Deep Neural Networks) on different types of viewing spaces discussed above. In each viewing space, we can show both the morphing surfaces as well as the different types of subspace projections discussed in Section 4.3. Please note that the color maps in different images are different because of the large differences in value ranges. A uniform color map would not be effective as we would see very little color contrast in each visualization image.

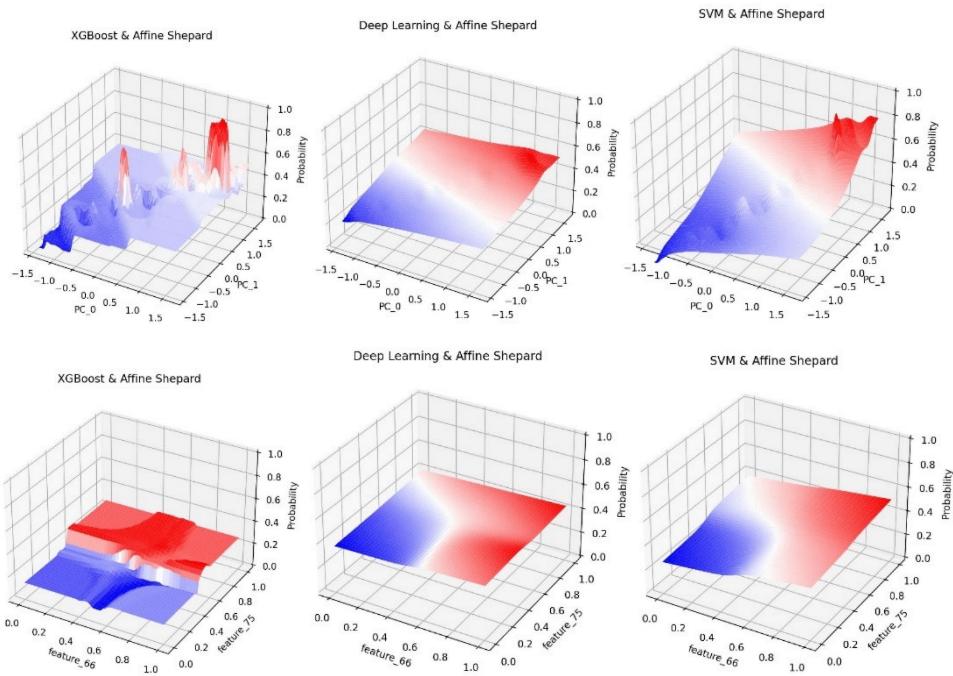
#### 5.3.2. Morphing Surfaces

Morphing surfaces are special cross-sections of the model space that interpolate the training samples. The two interpolation methods perform similarly. So, to save space, we will only show affine Shepard interpolation results. Figure 3 shows the morphing surfaces on the two common feature spaces: Age-BMI, and Education-MMSE for ADNI dataset. Figure 4 shows the morphing surfaces on the PCA space and a node degree feature space for ADNI dataset. Figure 5 shows the morphing surfaces on two 2D feature spaces for the Home Credit dataset.

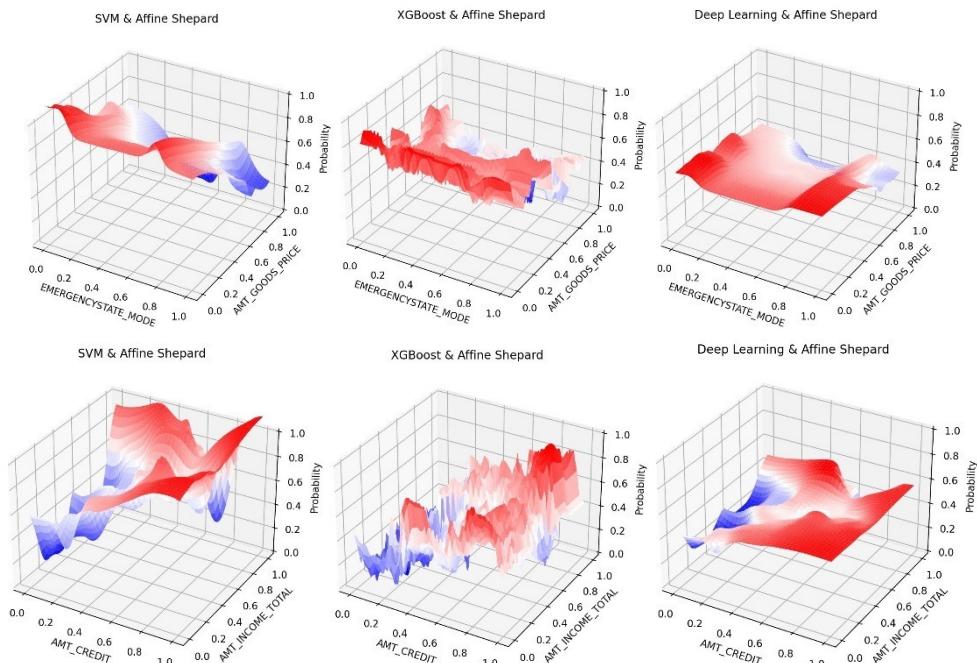




**Figure 3.** Morphing surfaces on Age-BMI and MMSE-Education using ADNI dataset.



**Figure 4.** Morphing surfaces on PCA space and a node degree feature space using ADNI dataset.



**Figure 5.** Morphing surfaces using Home Credit dataset.

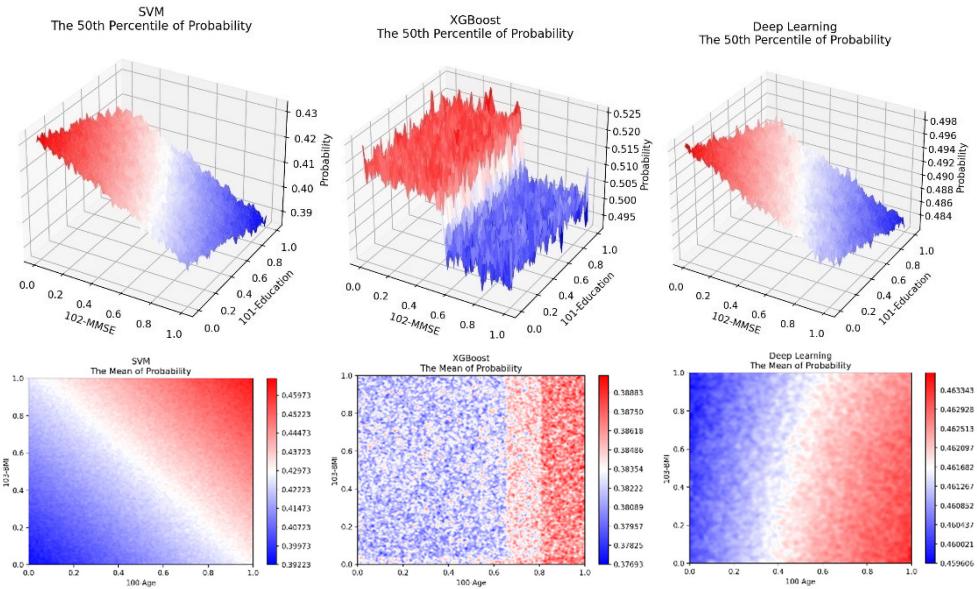
It is interesting to see that SVM and Deep Learning generate much smoother surfaces than XGBoost. This is somewhat understandable because of the ways these algorithms construct their models: SVM uses hyperplanes, and Deep Neural Networks mostly uses radial-basis functions to generate smooth approximations. But XGBoost is a decision tree styled process which tends to create discrete decision paths with sharp and block-shaped boundaries. This trend is particularly obvious in Figure 4 and Figure 5. Based on Figure 3 results, it appears that people in the late 50th with low BMI and people in the 80th with high BMI have higher risk of AD. We also see that education level does not seem to play a major role, but MMSE score is clearly a strong indicator of AD risk.

In Figure 5, we also see that the loan default risk is greater for high income and lower income borrowers outside the normal income range. This is consistent with some of the financial analysis reports for this kind of loans which is good validation case for our visualization even though it is a somewhat counter-intuitive. The results from Figure 5 also show that the home condition does not play a role in default risk. But interestingly loans for purchasing more expensive goods indicate lower risk of default.

### 5.3.3. Subspace Projection

Subspace projection takes random samples in each subspace, and then visualizes surfaces or heatmap images that represent different levels (percentiles) of the value ranges, or different types of subsets or statistical values of the samples.

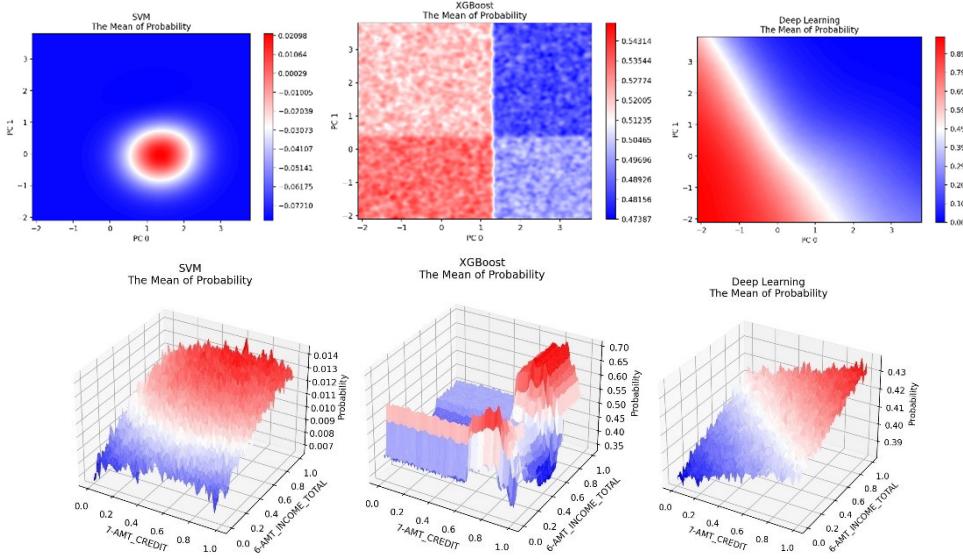
Figure 6 shows several 50 percentile value surfaces on MMSE-Education space and average or mean value heatmap images on Age-BMI space using the ADNI dataset. We find that surfaces are generally more visually informative, but sometimes heatmap images can also be very effective. These results show several interesting findings. (1) AD risk is higher for people with low MMSE scores. (2) People with higher level education will do slightly better in lowering AD risks. (3) Both older age and higher BMI level are risk factors for AD. (4) XGBoost algorithm exhibit block-like visual pattern while SVM and Deep Neural Networks have more gradual change in classification probabilities.



**Figure 6.** Some examples of 50 percentile surfaces and average or mean value heatmap images using the ADNI dataset.

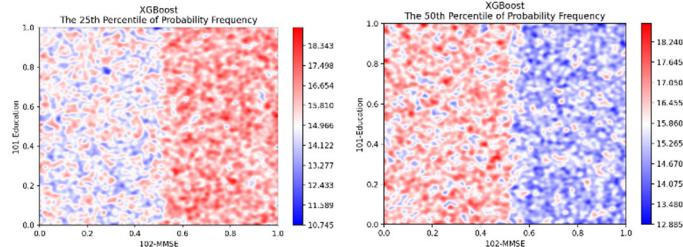
Figure 7 shows some examples of average or mean value visualizations for the Home Credit dataset. The PCA space images are very smooth for SVM and Deep Neural Networks. This is perhaps because the two models made some simple approximations (hence, the low accuracies) which showed up on the most representative PCA spaces. XGBoost again shows a strong block pattern. The

visualizations also show that higher loan amount leads higher risk of default. The XGBoost result also shows that very high and very low income levels lead to higher risks for default (consistent with our earlier observations). But SVM and Deep Neural Network did not capture the lower income portion which is likely because they have lower accuracies with this dataset. This also suggests that visualizing ML models from different algorithms may help us identify potential errors in some of the models.



**Figure 7.** Some examples of average value visualizations on a PCA space and a Credit-Income space using the Home Credit dataset.

Figure 8 shows the 25% and 50% cross-sections of the histogram distributions over the range of predicted values. Here we see that XGBoost have more low (25%) probability values in higher MMSE score area, but more 50% probability values in low MMSE score area, indicating that the probability of AD risk increases as MMSE score decreases.



**Figure 8.** Cross-sections at 25% and 50% for the histogram volumes on the MMSE-Education space using the ADNI dataset.

## 6. Conclusions

We have presented a general framework for visualizing supervised single valued machine learning models. We treat the ML models as continuous feature space functions with training samples. While visualization of machine learning processes is important for users to understand the decision-making process, it is often as important to provide a visual representation of the entire model to gain a high-level understanding about how the model behaves in different viewing spaces. Our approach differs from traditional higher dimensional data visualization as we aim to represent the global shape information of the model which is considered a manifold in a high-dimensional space. The morphing-based technique also considers the importance of the training samples. Morphing from a high-dimensional space to a lower-dimensional space is an interesting phenomenon that should be further explored. In addition, this type of model visualization can be an essential component for visual interactions in an interactive machine learning system or human-in-the-loop AI system. More specifically, model visualization can be used as an interface for users to

decide what actions need to be taken to incrementally improve the model, for example, by adding additional training samples.

In the future, we would like to further improve and extend this technique, and to develop a software solution that can be plugged in to some existing machine learning tools. We also want to explore the possibility of integrating this visualization approach in some machine learning assisted visual analytics systems to enhance their functionalities and analysis power. We are also interested in exploring the theoretical and practical aspects of other projection and rendering techniques, such as nonlinear projections, and blending techniques for subspace projection to produce richer visual perceptions in high dimensional space.

## References

1. Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* 2018; 6: 52138–52160
2. Angelos Chatzimparmpas, Rafael M. Martins, Ilir Jusufi, and Andreas Kerren. A survey of surveys on the use of visualization for interpreting machine learning models. *Information Visualization*. Volume 19, Issue 3, July 2020, Pages 207-233.
3. Seifert C, Aamir A, Balagopalan A, et al. Visualizations of deep neural networks in computer vision: a survey. In: Cerquitelli T, Quercia D, Pasquale F (eds) *Transparent data mining for big and small data*. Cham: Springer, 2017, pp. 123–144.
4. Liu S, Maljovec D, Wang B, et al. Visualizing high-dimensional data: advances in the past decade. *IEEE T Vis Comput Gr* 2017; 23(3): 1249–1268
5. Kaufman, A. (1992). Fundamentals of Volume Visualization. In: Kunii, T.L. (eds) *Visual Computing*. CG International Series. Springer, Tokyo. [https://doi.org/10.1007/978-4-431-68204-2\\_16](https://doi.org/10.1007/978-4-431-68204-2_16).
6. Sacha D, Sedlmair M, Zhang L, et al. What you see is what you can change: human-centered machine learning by interactive visualization. *Neurocomputing* 2017; 268: 164–175.
7. Ware, M., Frank, E., Holmes, G., Hall, M. and Witten, I.H., 2001. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3), pp.281-292.
8. H. Li, S. Fang, S. Mukhopadhyay, A. J. Saykin and L. Shen, "Interactive Machine Learning by Visualization: A Small Data Solution," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 3513-3521, doi: 10.1109/BigData.2018.8621952
9. Hao Wang, Snehasis Mukhopadhyay, Yunyu Xiao and Shaoften Fang. An Interactive Approach to Bias Mitigation in Machine Learning. *IEEE 20th International Conference on Cognitive Informatics and Cognitive Computing (ICCI\*CC'21)*, 2021
10. Huang Li, Shaoften Fang, Joaquin Goni, Andrew Saykin and Li Shen. Interactive Visualization of Deep Learning for 3D Brain Data Analysis. *IEEE 20th International Conference on Cognitive Informatics and Cognitive Computing (ICCI\*CC'21)*, 2021
11. Endert A, Ribarsky W, Turkay C, et al. The state of the art in integrating machine learning into visual analytics. *Comput Graph Forum* 2017; 36(8): 458–486
12. May T, Bannach A, Davey J, et al. Guiding feature subset selection with an interactive visualization. In: *Proceedings of the 2011 IEEE conference on visual analytics science and technology (VAST)*, Providence, RI, 23–28 October 2011, pp. 111–120. New York: IEEE.
13. Paiva, J.G., Florian, L., Pedrini, H., Telles, G., Minghim, R., 2011. Improved similarity trees and their application to visual data classification. *IEEE TVCG* 17 (12), 2459–2468
14. Jing Xia, Wei Chen, Yumeng Hou, Wanqi Hu, Xinxin Huang, David S. Ebert. DimScanner: A Relation-based Visual Exploration Approach Towards Data Dimension Inspection. *VAST 2016*.
15. Shixia Liu, Xiting Wang, Mengchen Liu, Jun Zhu. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics* 1 (2017) 48–56.
16. Zahavy, T., Ben-Zrihem, N., Mannor, S. 2016. Graying the black box: Understanding dqns. In: *ICML* pp. 1899–1908.
17. Rauber, P.E., Fadel, S., Falcao, A., Telea, A., 2017. Visualizing the hidden activity of artificial neural networks. *IEEE TVCG* 23 (1), 101–110.
18. Tzeng, F.Y., Ma, K.L. 2005. Opening the black box - data driven visualization of neural networks. In: *IEEE Visualization*, pp. 383–390. <http://dx.doi.org/10.1109/VISUAL.2005.1532820>.
19. Harley, A.W., 2015. An interactive node-link visualization of convolutional neural networks. In: *International Symposium on Visual Computing*. Springer, pp. 867–877.
20. Streeter, M.J., Ward, M.O., Alvarez, S.A., 2001. Nvis: An interactive visualization tool for neural networks.
21. Liu, M., Shi, J., Li, Z., Li, C., Zhu, J.J.H., Liu, S., 2017. Towards better analysis of deep convolutional neural networks. *IEEE TVCG* 23 (1), 91–100. <http://dx.doi.org/10>.

22. Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding Neural Networks Through Deep Visualization. ICML Workshop on Deep Learning, 2015.
23. Luisa M Zintgraf, Taco S Cohen, Tameem Adel, Max Welling. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. International Conference on Learning Representations (ICLR) 2017.
24. SeungJin Lim. A Light-Weight Visualization Tool for Support Vector Machines. 25th International Workshop on Database and Expert Systems Applications, 2014.
25. Lutz Hamel, Visualization of Support Vector Machines with Unsupervised Learning, IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, 2006.
26. Wang, J; Fang, S; Li, H; Goni, J; Saykin, AJ; Shen, L. Multigraph Visualization for Feature Classification of Brain Network Data. EuroVis Workshop on Visual Analytics (EuroVA), pp.61-65, 2016.
27. Ren, D., Amershi, S., Lee, B., Suh, J., Williams, J.D., 2017. Squares: Supporting interactive performance analysis for multiclass classifiers. IEEE TVCG 23 (1), 61–70.
28. Alsallakh, B., Hanbury, A., Hauser, H., Miksch, S., Rauber, A., 2014. Visual methods for analyzing probabilistic classification data. IEEE TVCG 20 (12), 1703–1712.
29. Chuang, J., Gupta, S., Manning, C.D., Heer, J. 2013. Topic model diagnostics: Assessing domain relevance via topical alignment. In: ICML, pp. 612–620.
30. Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li and David S. Ebert. Manifold: A Model-Agnostic Framework for Interpretation and Diagnosis of Machine Learning Models. IEEE Transactions on Visualization and Computer Graphics, 25(1), 2019, pp 364 – 373.
31. Beier, T. and Neely, S. (1992), Feature-based image metamorphosis, Computer Graphics, SIGGRAPH'92 (26) 2, 35–42.
32. Shiaofen Fang, R. Srinivasan, S. Huang and Raghu Raghavan. Deformable Volume Rendering by 3D Texture Mapping and Octree Encoding. Proc. of IEEE Visualization'96, pp. 73-80, Oct. San Francisco, CA.
33. Shiaofen Fang, R. Srinivasan, Raghu Raghavan and Joan Richtsmeier. Volume Morphing and Rendering - - An Integrated Approach. Journal of Computer Aided Geometric Design, 17(1):59-81, January, 2000.
34. Cook, P., Bai, Y., Nedjati-Gilani, S., Seunarine, K., Hall, M., Parker, G. and Alexander, D. Camino: open-source diffusion-mri reconstruction and processing. 14th Scientific Meeting of the International Society for Magnetic Resonance in Medicine, Vol. 2759, Seattle WA, USA, 2006.
35. Liu, Shusen & Wang, B. & J. Thiagarajan, Jayaraman & Bremer, Peer-Timo & Pascucci, Valerio. (2015). Visual Exploration of High-Dimensional Data through Subspace Analysis and Dynamic Projections. Computer Graphics Forum. 34. 10.1111/cgf.12639.
36. Chiu, Chia-Hsing, et al. "Human-in-the-loop differential subspace search in high-dimensional latent space." ACM Transactions on Graphics (TOG) 39.4 (2020): 85-1.
37. Gerber, Samuel, et al. "Visual exploration of high dimensional scalar functions." IEEE transactions on visualization and computer graphics 16.6 (2010): 1271-1280.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.