SUBPLEX: Towards a Better Understanding of Black Box Model Explanations at the Subpopulation Level

GROMIT YEUK-YIN CHAN and JUN YUAN, New York University
KYLE OVERTON, BRIAN BARR, and KIM REES, Capital One
LUIS GUSTAVO NONATO, Universidade de São Paulo
ENRICO BERTINI and CLÁUDIO T. SILVA, New York University

Understanding the interpretation of machine learning (ML) models has been of paramount importance when making decisions with societal impacts such as transport control, financial activities, and medical diagnosis. While current model interpretation methodologies focus on using locally linear functions to approximate the models or creating self-explanatory models that give explanations to each input instance, they do not focus on model interpretation at the subpopulation level, which is the understanding of model interpretations across different subset aggregations in a dataset. To address the challenges of providing explanations of an ML model across the whole dataset, we propose SUBPLEX, a visual analytics system to help users understand black-box model explanations with subpopulation visual analysis. SUBPLEX is designed through an iterative design process with machine learning researchers to address three usage scenarios of real-life machine learning tasks: model debugging, feature selection, and bias detection. The system applies novel subpopulation analysis on ML model explanations and interactive visualization to explore the explanations on a dataset with different levels of granularity. Based on the system, we conduct user evaluation to assess how understanding the interpretation at a subpopulation level influences the sense-making process of interpreting ML models from a user’s perspective. Our results suggest that by providing model explanations for different groups of data, SUBPLEX encourages users to generate more ingenious ideas to enrich the interpretations. It also helps users to acquire a tight integration between programming workflow and visual analytics workflow. Last but not least, we summarize the considerations observed in applying visualization to machine learning interpretations.

CCS Concepts: • Human-centered computing → Information Visualization; Visualization design and evaluation methods.

Additional Key Words and Phrases: explainable AI, visual analytics

ACM Reference Format:
Gromit Yeuk-Yin Chan, Jun Yuan, Kyle Overton, Brian Barr, Kim Rees, Luis Gustavo Nonato, Enrico Bertini, and Cláudio T. Silva. 2018. SUBPLEX: Towards a Better Understanding of Black Box Model Explanations at the Subpopulation Level. J. ACM 37, 4, Article 111 (August 2018), 28 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

With the advance of computing power, machine learning (ML) produces accurate prediction models that can be applied to address important societal problems like financial fraud detection [35], drug discovery [47], and natural disease prediction [74]. On the one hand, people aim at training models
with high accuracy that often are achieved by complex decision boundaries that capture subtle varieties from the data. On the other hand, stakeholders and end-users expect scientifically rigorous explanations from the models that provide understanding, protect the safety, and ensure ethics [20]. To balance the development between model sophistication and human understanding, a burgeoning research field of explainable artificial intelligence (XAI) has arisen. The general goal of XAI is to develop human-understandable explanations of what a model has learned.

In general, the two main scopes of understanding how model works are general overviews of model behavior (global explanations) and precise decision details of each instance (local explanations). These explanation models target only the features in the dataset. Thus they are consistent with different types of tasks such as classification, regression, language translation, and object recognition. Global explanations are mechanisms that describe how a model works overall using simpler logic or approximations such as rules [46, 48] or multiple linear models [15, 89]. Local explanations focus on generating sparse interpretable vectors like prototypes [17, 49], concepts [40], or feature weights [73, 83] for each input data. Both play an essential role in model interpretability and complement each other. For example, users leverage global explanation to evaluate whether the model achieves some general goals like learning the hierarchy of different classes [8] from the dataset. Afterward, they may require some sanity check on individual data to verify that their understanding is consistent with the internal structures of the model [40].

As a result, there is a need to take the granularity of explanations to an appropriate subpopulation level. A subpopulation, in other words, the subset of instances’ explanations in the dataset, provides an overview of decision characteristics from different major parts of the data. It acts as a bridge between an overly coarse global view and extremely detailed information of a single instance. Thus, exploring the subpopulation allows the users to find a proper balance in the data exploration process. At the same time, the challenge of understanding model explanations through subpopulation is straightforward – how to find the best partition among a dataset of instances’ explanations. Like clustering, there are many ways to cluster a dataset. Finding the best subpopulation, in other words, a subpopulation analysis, is a computation challenging and human-centered question.

Furthermore, to realize the potential of model interpretability for end-users, we need the explanations to be provided in an integrated platform and in a human-centric way. Recently, information visualization has been receiving much attention as a medium for model explanations [30], and different visual analytics systems have been developed to address this challenge [29, 37, 54, 63]. Intuitively, visualization enhances model interpretability since graphical representations have been shown useful to communicate complex statistics [87]. Using projections, clustering, and interactions [39], visual analytics allow users to interpret large amounts of information, revealing intrinsic global patterns while maintaining the ability to explore details. Thus, combining visual analytics and model explanation techniques provides a promising area of improving machine learning model interpretability.

In this work, we take the problem of understanding model interpretability as a subpopulation analysis of local explanations. If we treat the local explanation for each input data as the target, we aim at visualizing and displaying the similarity and dissimilarity of all local explanations together, which allows us to discover the main decision rationales (i.e., clusters) as well as more detailed considerations (e.g., outliers). Also, as these explanation methods work for a versatile range of machine learning applications, we are interested in the potential of analyzing explanations as a standalone goal. In this way, the output can inherit the flexibility of model explanations and be embedded in a wider machine learning process. Having this objective in mind, we designed SUBPLEX, a visual analytics system that visualizes machine learning model explanations at a subpopulation level. We also develop it as a widget in the computational notebook to study the opportunity of analyzing model explanations as a standalone task. Working as a team of

J. ACM, Vol. 37, No. 4, Article 111. Publication date: August 2018.
5 visualization researchers and 3 data scientists, we combine the concepts of subpopulation analysis in visualization and real industrial tasks on model interpretability for model developers to induce the workflow of model interpretation from local explanations at scale. In short, our contributions include:

1. An overview of combining subpopulation analysis and machine learning explanation into a visualization system, including a discussion of tasks, techniques, and visual design.
2. A discussion of user evaluation on the workflow of model interpretation from understanding local explanations from the dataset.

2 RELATED WORK

In this section, we first discuss how visualizations are applied in the area of machine learning model explanations. We then explain the motivations of subpopulation analysis for those explanations. Finally, we present the related critical approaches in human-computer interaction (HCI) and machine learning communities. We first present the critical approaches for model interpretation in machine learning communities and how visual analytic is applied in this area. We then discuss the human factors in interpretable machine learning. Finally, we explain the motivations of subpopulation analysis and how this can help with model interpretations.

2.1 Visualization for Model Explanation

With the increase in the complexity of machine learning models in recent years, the interpretation of machine learning models has been highly valued. Machine learning and visualization communities have been working long on the explanation of machine learning models to improve fairness [59], support debugging [4, 55, 70], comparisons [95], and gain trust from end-users [56].

There are two distinct categories of model explanation approaches: “white-box” approaches, and “black-box” approaches [66]. White-box, or intrinsic, approaches tend to restrict the internal logic and structure of a model so that human observers can understand the logic of why a decision is made. Black-box, or post-hoc, approaches try to explain how the inputs related to the output without showing the internal working mechanism of a model.

White-box approaches are applicable for those intrinsically interpretable models, such as decision trees, rule-based models, and linear models. For example, a gallery of tree visualization can be found at treevis.net[78] for visualizing decision trees. BOOSTVis[55] helps model diagnosis during the training process of tree boosting. Rule-based models are composed of logical representations and can be expressed in a list of IF-THEN or IF-THEN-ELSE statements. People can get insights into a linear model by using projection-based methods[14].

Black-box models, whose internals are generally opaque and uninterpretable, cannot be interpreted by using a white-box approach. Although these models are known to provide better performance in many cases[27], using black-box models in high-stakes scenarios may result in increased risk[65], lower trust, and limited adoption due to lack of interpretability [56]. Thus, there is a surge of interest in the interpretation of black-box models (check the survey [27] for more details). In contrast to the white-box approaches, black-box model explanations focus more on the relationship between input and output without looking at the internal structure of the model. In our work, we leverage model diagnosis functions for data scientists who work with black-box models and need to understand the model with domain knowledge so that black-box approaches are adapted.

In general, black box explanations can be divided into three classes as categorized in this survey [27]: Model Explanation when the explanation involves the whole logic of the model, Outcome Explanation to explain why a specific decision is made for a given object, and Model Inspection
focusing on providing a representation (visual or textual) for understanding some specific property of the black box model or of its predictions when input changing. To provide a \textit{model explanation} for a global understanding of the model, an interpretable surrogate model needs to be trained to approximate a black-box model. This model should be both able to mimic the behavior of the black box, and it should also be understandable by humans. However, the complexity of a surrogate model increases in order to better approximate a black box. Ming \textit{et al.}\cite{62} presents the trade-off between model complexity (related to interpretability) and the fidelity to the original model. To work with the real input and model output and keep the explanation understandable, we try to include outcome explanations and model inspection in this project.

Instead of understanding a surrogate model, \textit{outcome explanations} and \textit{model inspection} work on the original data space. For decision-makers, the explanation of a given case is helpful when making decisions. To this end, interactive visual systems are proposed to understand the specific explanations more effectively and efficiently. For example, Krause \textit{et al.}\cite{41} leverages instance-level explanation, measures of local feature relevance that explain single instances in an interactive visual system to help domain experts to investigate, understand, and diagnose the model decisions. And in many recent applications\cite{43, 73}, both \textit{outcome explanations} and \textit{model inspection} are integrated with visualization to help users understand model decisions. Prospector\cite{43} provides interactive partial dependence diagnostics to present how features affect the prediction overall, together with localized inspection for a better understanding of how and why specific data points are predicted as they are. LIME\cite{73} proposed an algorithm of finding out the attributions of different features by adding perturbation to the original input and then highlight the attributions relevant to the prediction in visualization.

For model inspection of complex models, such as deep neural networks, visualizations can aid developers in understanding the internal structures of the model. For example, TensorFlow\cite{93} visualizes the underlying dataflow graph of a deep learning model. Liu \textit{et al.}\cite{53} used a hybrid visualization that embeds debugging information in the neural network visualization to help experts understand, diagnose, and refine deep CNNs. Tzeng \textit{et al.}\cite{88} introduces the visualization of weights on neural networks with a single instance or a set of data instances to gain more understanding and confidence in using artificial neural networks. And ActiVis\cite{36} not only visualizes the internal structure of neural network models but supports model exploration at both the instance- and subset-level.

\textbf{Take-away}: Visualizations are applied a lot in the area of machine learning explanation to assist humans to grasp a better understanding of a machine learning model. In our work, we treat a model to be explained as a black box. Moreover, according to the task analysis in the next section, our research focuses more on explaining the model from the data space, which falls into the categories of model inspection and outcome explanation in terms of explaining a black box. Furthermore, to inspect the model behaviors, explanations from multiple granularity levels are required, such as an instance-level explanation for a single model decision, and a subpopulation-level explanation for a group of instances that the model makes decisions for similar reasons. We include more details in a later section, discussing how subpopulation analysis with the help of visualization can assist model interpretation.

\section{Human Factors in Interpretable Machine Learning}

Since the end-users of machine learning interpretations are the humans themselves, it is crucial to address real-world user needs for understanding AI and generate human-friendly explanations for users ranging from model developers to domain experts and decision-makers.

Lipton proposes an overview of machine learning model interpretability \cite{52}, where the author summarizes the properties of interpretable models and addresses that humans need model interpretation
so that they can build trust in the model, make more informative, fair and ethical decisions. From the perspective of human-computer interaction, there are considerations more than reasoning to consider a model interpretable and useful. The awareness of reasoning [72], trust [26, 73], alignment with user expectation [22], justice [9], contrastive reasoning [51], and human-in-a-loop analysis [45] are all possible factors to affect the willingness of users to apply machine learning models to the application scenario.

Human decision making, the subsequent step of model understanding, is also an essential factor for generating interpretable models. Whether users are willing to make decisions that are based on the models depends on the model’s accuracy [5, 24], its variances on different input and output [84], and the availability of performance reports [86].

Although there are many explainable AI (XAI) algorithms are proposed as stated in the previous subsection, a recent research [50] reports the interview results with practitioners from the industry revealing that it remains a challenge, for now, to create explainable AI products because of the variance of user needs for explainability, discrepancies between algorithmic explanations and human explanations and a lack of support for design practices. Furthermore, the HCI community has also called for interdisciplinary collaboration [1] and user-centered approaches to explainability [90] to bridge the gap between XAI algorithms and user needs for sufficient transparency.

**Take-away:** With the history of previous work into XAI, it is essential yet challenging to design an XAI product that addresses the real issues when explaining a machine learning model. In our work, we are going to present a hierarchical task analysis in the later section, which maps the design goals to multi-level tasks as well as how our designs evolve during the collaboration with data scientists from the industry.

### 2.3 Subpopulation Visualization

A clustering algorithm helps the dataset to discover groups of similar objects. Clustering has become a popular unsupervised learning method [85] typically used early in the process of exploratory data analysis. Cluster analysis has a heuristic nature that encourages the exploration of data [21]. Inspired by this, we believe that data analysts can benefit from generating hypotheses at the subpopulation level. Dimension reduction algorithms and clustering algorithms are both frequently used techniques in visual analytics. Both categories of techniques assist analysts in performing related tasks regarding the similarity of observations and finding groups in datasets [91]. In terms of model inspection and outcome explanation, the exploration of subpopulation presents the groups that can be explained by similar reasons, as well as outliers where the model has abnormal behavior patterns with them. In the following subsections, we are going to explain the usage of clustering visualization and dimensionality reduction methods which assist subpopulation analysis.

#### 2.3.1 Visualizing Clusters.

In general, there are three categories of visualization of clusters: (1) visualizing membership of clusters, focusing on presenting the groups that data instances belong to; (2) visualizing the content of clusters, aiming at demonstrating the feature values or properties of data instances in a cluster; (3) cluster optimization, where the visual system enables users to modify the membership of instances to reach a customized clustering result.

Saket et al. studied three options for encoding group membership: nodes with colors of cluster membership, nodes with cluster colors and links, as well as colored space-filling regions. Jianu et al. [33] further explored the options of Linsets [2], GMap [25], and BubbleSets [19]. The visualization of clusters or groups provides a straightforward way of showing data distribution. A recent application of clustering for explainable machine learning is CNN2DT [32], where bubble sets are used to highlight the regions of neurons in CNN with the same label.
In recent years, many interactive systems also include the visualization of cluster content to assist users to explore the clustering results. For example, a heat map, as applied in Hierarchical Clustering Explorer (HCE) [81] is used to show the overall feature values in clusters. Parallel coordinates [31] is another type of chart that is widely used for multidimensional data. Its application in ClusterVision [44] enables data distribution overview and useful cluster comparison. However, the usage of parallel coordinates can be cluttered when too much data is being visualized [7, 94].

Another type of visual systems for clustering is designed for cluster optimization. For example, Packer et al. [69] use heuristics to suggest interesting algorithmic settings for exploration. SOM-Flow [76] enables further data partitions for existing clustering output. Moreover, ClusterVision [44] can retrieve new clustering results recommended based on users’ input.

2.3.2 Dimensionality Reduction in Visualization.
A recent work [67] provides a survey of Multidimensional Projections (MDP) methods, properties, errors, and tasks. MDP algorithms such as T-SNE [57], Umap [60], LAMP [34], PCA [92], MDS [10] are widely used in the visualization communities. As for the visual representation of MDP, most dimension reduction algorithm outputs are shown in scatterplots or node-link diagrams [91]. For instance, Andromeda [80] integrates a 2D projection view to support communication between a user and high-dimensional data analysis. Kogan introduces Star Coordinates [38] that arranges coordinates on a circle sharing the same origin at the center for cluster discovery and multifactor analysis tasks. Besides numerical data, text data [3, 12], and image data [58] can also be encoded in a scatterplot using MDP techniques.

With only the layout resulting from an MDP mapping, we can get a basic point cloud where groups and neighborhoods are indicative of similarity among the involved instances. However, content-based enrichment techniques that build upon the proximity of similar instances in the visual space can be exploited to depict additional information associated with particular instances or groups. For example, Facetatlas [13] exploits a cluster-based enrichment for to highlight the clusters in a projection view. Though initially clustering and dimensionality reduction algorithms are used independently, recent works have incorporated algorithms from each family into the same visualization systems. As pointed out in this survey [91], there can be six different options for pipelines depicting combinations of dimension reduction algorithms and clustering algorithms. In our work, we try to achieve our design goals by integrating cluster analysis on multidimensional data so that multidimensional projection with cluster-based enrichment is considered in our visual designs.

Take-away: Clustering and dimensionality reduction algorithms are widely used for subpopulation analysis, which assists interactive model inspection and outcome explanation. Inspired by this, our work provides an interactive approach to subpopulation level model exploration to help users grasp a better interpretation of the model and data.

3 DESIGN PROCESS AND RATIONALE

3.1 Addressing Real World Goals to Understand Model Interpretability
Interpretability is a vague concept that could be either as general as understanding a logical reasoning process or as niche as developing designs and tools that solve a real world problem that requires experts to understand black-box models for decision making. Our motivation for contributing to the current literature comes from a year-long collaboration with a retail finance institution in which we have implemented a model explanation interface for the credit scoring system by exchanging ideas between the finance experts and visualization researchers. The experts are mainly model developers who have sufficient knowledge of the data and the models. Thus,
Fig. 1. Hierarchical task abstraction of three model interpretability goals from our domain experts. Each goal has its own hierarchy of tasks (G1-3). For each hierarchy, the box represents a task or subtask and each level of hierarchy has a plan. The line under the box means a termination. The highlighted green text represents the task abstraction derived from the tasks.
their motivations of using model explanation methods are to leverage the exploration of important features to address the interpretability goals. By addressing the everyday model explanation tasks in the financial operations, we developed a new perspective of model interpretation through careful consideration of subpopulation analysis and visual design. While there are no guarantees of completeness, our system design and design rationale are based on the goals of understanding black box model behavior in the credit score system. Each goal is provided with an example of a model interpretability question, which is related to decision making in the financial operations.

**G.1 How does the model explain different groups of customers?**

In a retail financial institution, practitioners aim at developing models that can be used for a considerably large amount of customers to improve efficiency while ensuring that it provides a degree of discriminatory power to different populations so that the model is not over-generalized with simple rules. For example, an ideal model should learn to use different features on customers with different demographics while maintaining the use of default rates on the general public.

**G.2 What does the model learn after removing bias features?**

The term *bias* here does not only mean features related to machine learning fairness but also the dominating features that may decrease the diversity of granting credits to different users. For example, experts would like to see what are the next level influential features that affect credit scoring without considering how many mortgages the customer owns so that more exciting features can be discovered for future financial products.

**G.3 Are the model’s predictions affected by spurious information?**

This is a model debugging problem that developers need to consider very carefully when they put the model into production. A typical way to examine this in practice is to include some false or random variables in the model and see how are the populations be affected by the addition. For example, the developers would like to know can the population with a low default rate receives a good credit score by increasing their length of credit history? If so, they may be a chance to “cheat” the model with adversarial attacks.

### 3.2 Breaking Down the Goals into Tasks

The above three goals, while providing the motivations to develop a visual analytics solution, do not explicitly invoke design rationale for our system design. Therefore, it is important to extract the low-level details and actions from these three high-level goals to address the key needs to develop a visual analytics system. These details can be analyzed and mapped to a system-level task requirement. To acquire the low-level tasks, we examine the workflow of our expert through their analysis in the Jupyter notebook. Jupyter notebook is a mainstream data analytics platform that allows data scientists to execute Python scripts to model data and return results in a list of sequential cells. Thus, we studied the notebooks from five data scientists working on these goals and extracted the workflow of the data analysis through browsing the data operations in each cell in the notebook sequentially.

Once we obtain the workflow of data operations to address those goals, we formulate the whole analytics workflow as an exclusive and exhaustive Hierarchical Task Abstraction (HTA) [6]. HTA is a popular approach in the HCI community to summarize the tasks conducted by the end-users. It incorporates a set of goals and low-level tasks as a hierarchy to help researchers understand both the necessary tasks and the goals and process. Recently, it has been used by design studies in visual analytics application development [16, 96] as well.

The breakdown of the goals can be seen in Figure 1. In general, each goal can be achieved by around three to five main themes of data analysis, which consists of summarizing a model’s decision
rationale, selecting an interesting portion of instances and features, and applying further data operations. By grouping the lowest level tasks among the three goals, we summarize the overall task requirement in Figure 1:

**T.1 Interactive clustering to generate subpopulation of local explanations.** All of the three use cases require an overview of instances’ explanations to understand the model’s decision rationale. Therefore, a clustering result of instances based on their similarity of explanation helps users to identify decision paths on the major population as well as the outliers in the dataset. While an initial partition can be generated by automated algorithms to kick start the subpopulation analysis, users also need to refine the results such as merging or splitting the clusters so that the groups of explanations suit their analytics purposes. For example, for model debugging (G.3), the purpose of clustering is to isolate the instances of which the model relies heavily on spurious information to make decisions. Tailoring the clustering results thus is needed to provide the desired data for further analysis. In other words, users combine data mining algorithms and interactions to address the tasks.

**T.2 Visual analysis of explanation partitions.** Once the subpopulation of local attributions is finalized, users need to inspect the characteristics of each subpopulation to decide which features or instances should be focused on further data analysis or model refinement. We observe that using basic plotting libraries in Jupyter notebook, our expert still applies a workflow of visual analysis: they first inspect an overview of feature importance over the dataset, then search for an interesting subset of data to focus on its details such as the size of subset and their most-used features. Thus, users require the system to display overview as well as detail-on-demand to identify a more focused group of data and features for further analysis.

**T.3 Seamless integration of data analysis pipeline and infrastructure.** As the subpopulation analysis is a part of the whole model interpretation workflow (i.e., the middle between data preprocessing and data communication or model refinement), it is essential to integrate the whole stage of analysis into the current programming infrastructure so that we can reduce the overhead among switching different platforms or storing many intermediate files. The whole subpopulation analysis should take the input inside the Jupyter notebook and output results to the notebook. In such a case, users can assess the results and save the input as variables to recycle written codes to conduct iterative analysis and different trial and error experiments to facilitate creativity.

### 3.3 Design Rationale for Visual Analytics

Given a set of tasks we summarized in **T.1-3** and the exchange of ideas with our domain experts, we formulate the design rationale of our visual analytics system:

**R.1 Visual and interactive clustering of local explanations.** The system should provide ways to cluster the instance explanations from the trained model. Also, it should provide flexibility for the user to adjust and refine the results of the clustering to create partitions that suit various objectives.

**R.2 Focus on explanations in the whole interface.** Since the local explanation models work for a variety of tasks, including but not limited to classification, translation, and object detection. Our whole framework and interface should focus on the data generated by the explanation method to achieve generic usage.

**R.3 Display of similarity and difference among instances and general as well as outlying behavior.** For data with the same group, the model explains them similarly. Otherwise, there are differences in terms of the attribution values. At the same time, the size of groups also
indicates that the instances represent general or outlying behavior. The system should display these properties.

**R.4 Focus on data variety but not design variety.** Data scientists often use a well-known set of visual encodings to display the outcomes of machine learning models. Our solution should respect their mental model and provide the desired workflow and interactions to address the problems.

**R.5 Widget based system implementation leveraging the infrastructure and utility in Jupyter notebook.** Since the workflow of visual exploration is in between data operations, which heavily use multiple Python libraries such as scikit-learn and Tensorflow, our system should be embedded in the same environment. The interface should take inputs not only from user interactions but also provides APIs for querying and manipulating data in the interface.

To maximize the following objectives, we employ the subpopulation visual analysis, which is common in analyzing the similarity of observations and finding groups in datasets [91]. The visual analytics consists of two main components to facilitate the sensemaking process:

- **Partitional Clustering:** Partitioning the whole population into different clusters allows users to observe a clear split of data groups by their feature values. Subpopulations can be clearly defined by automated algorithms so that data characterized by different features and intrinsic decision-making processes in the models can be revealed by different clusters (T.2 and T.3).
- **Projection:** This allows the data to be spatially organized on display according to the similarity. Thus, community structure and outliers can be observed. Users can observe whether there are significant groups and whether data points are having much-deviated behavior compared with the majority of the population, which are useful for a general model understanding (T.1).

Nonetheless, such a form of visual analytics is not trivial, especially for the task requirements of model explanation and the data format of the explanation models, in which we are going to propose the methodology and visual design in the following sections.

### 4 SUBPOPULATION MODEL FOR BLACK BOX EXPLANATION

In this section, we describe the framework that we apply to produce the explanation subpopulation for visual analytics. We first explain the representation of local explanation for the input data. Then we describe the data model that takes these explanations to produce subpopulation analysis.

#### 4.1 Background of Local Explanation Models

We first give a background of the mainstream models that generate local explanations of a machine learning model’s decisions to a dataset. The popularity of giving local explanations, except applying logical models such as decision trees or rules, is because these methods provide an independent and highly customized explanation for each instance. When explanations do not aggregate into general decisions or rules, they become more faithful to the original model.

In general, to generate a local explanation for an instance, explanation algorithms usually seek one of the following approaches:

1. **Locality:** The algorithm searches the neighbors of an instance, then fits the subset to a linear model such that the higher the gradient of a feature in the linear model, the more important the feature is to the prediction of the selected instances.

2. **Perturbation:** Instead of using other instances to generate explanations, one can perturbate the values of its attributes and observe whether the output changes significantly. The sensitivity of
each feature implies that its value lies in the decision boundary of the machine learning model. Thus, a sensitive feature from perturbation has a high influential power on the instance.

(3) Backpropagation: Since complex models like neural networks contain series of propagation of weights from the input to the output neurons to produce predictions, one can invert the process to backpropagate the active neurons from the output to the input data locate the portion of original data that causes the neuron activations in the output. Such a portion implies the important features that explain the model’s decision.

4.2 Data Interpretation Representation

The first question of generating explanation is what constitutes an explanation for a data point; in other words, attribution, that a human can understand? Although there are no formal technique definitions for interpretability, the popular explanation models generate attribution in similar ways. Current models usually express the attribution for a data point as a sparse or skewed vector where each value inside the vector is a human-understandable object. For example, additive feature attribution methods like LIME [73], DeepLift [83], and GAM [28] output the explanation as a list of feature importances for each data (i.e., this data has these features), and prototype learning methods [49, 64] explain each data with a list of similarity with other data points (i.e., this data “looks” like that data). Therefore, we can define attribution \( a_i \) for each input point \( i \) as a set of real valued weights mapped to a feature space with \( m \) samples:

\[
a_i = \{w_1, w_2, \ldots, w_m\} \quad \forall w \in \mathbb{R}
\]  

where each weight \( w \) represents the attribution value for a feature. Although there does not exist any hard constraints when generating the attribution vector, the methods, in general, try to achieve the following objectives:

(1) Sparsity: The attribution vector should not contain many weights with high values (i.e., most of the \( w \) in Equation 1 are close to zero). This ensures that the data can be explained using a small set of features, taking human short term memory of a few items (e.g., not more than seven [61]) into account of interpretability.

(2) Diversity: As only a few items should be shown to explain a data point, it is also crucial to ensure that each feature shown should not be similar to each other. This objective often co-exists with sparsity as choosing the most distinctive and discriminative features results in a sparse, and thus a less redundant set of explanations.

4.3 Generating Attribution Subpopulation

Once the attribution for each data is generated, subpopulation can be discovered by clustering them by their similarity. The main challenge we need to address is how to compute the distances between the attribution vectors so that the clustering is accurate and efficient. Since the attribution is a sparse vector with many values (e.g., number of training samples in prototype learning), if we cluster the attributions with Euclidean distance, the clustering result will suffer from curse of dimensionality and be easily distorted by small perturbations. Also, the most efficient clustering algorithm (i.e., K-Means clustering) requires \( O(k \cdot i \cdot n \cdot d) \) time complexity, where \( k \) is the number of clusters, \( i \) is the number of iterations, \( n \) is the number of data points, and \( d \) is the number of dimensions. While the number of iterations \( i \) can be fine-tuned and computation for each \( n \) data points can be parallelized, if we do not control the dimension \( d \) within a small range, the running time would inhibit interactive analysis (R.1).

To prepare data to fit into the clustering algorithms more efficiently, we propose the use of Principal Component Analysis (PCA) to transform the sparse attribution vector into a low dimensional
Fig. 2. Distribution of attribution values on the synthetic dataset. The first half of the data (left) is explained by feature A and the second half (right) is explained by feature B, illustrated by distributions of higher values in the corresponding box plots.

Fig. 3. Accuracy and run time of clustering with the addition of sparse noise columns. The accuracy will decrease drastically and run time will increase when the attribution vector becomes sparser. With the transformation of PCA to the attribution vectors before the clustering, the accuracy will become more robust to noise and the computing time will be much faster.

vector that preserves as much information as possible by maximizing variance. Thus, the euclidean distance between these vectors will represent the cluster characteristics more significantly.

We now illustrate the effectiveness by conducting the following experiments with a synthetic dataset. The dataset $D$ first consists of two classes, two features (A and B), and 10000 points in total. The first half of the dataset is predictive by features A and the second half is predictive by feature B. This is achieved by assigning feature values in the following way:

$$D_{1,2,...,5000} = \begin{cases} 
\text{feature A} = \begin{cases} U[0,1] \text{ if class 1} \\
U[1,2] \text{ if class 2} 
\end{cases}, & \text{feature B} = U[1,2] 
\end{cases}$$
Fig. 4. System overview that integrates two coordinated views. (A) The **Projection View** displays all attribution vectors in a two-dimensional layout. (B) The **Subpopulation Details View** displays all feature attribution values among all subpopulations. (i) The bar charts visualize the average attribution values of a column within the same group, while (ii) the distribution chart superposes all grouped distributions of the attribution values of a column of each subpopulation.

We split the dataset into a train/test split of 80/20 and achieve a test accuracy of 99.95% with a random forest classifier. We run LIME with all the data and classifier, which generates the attribution vectors with the characteristics shown in Figure 2. Overall, data that are explained by a feature to a greater extent is assigned higher attribution values on the corresponding feature.

To illustrate the effect of noise and the effectiveness of our attribution transformation approach, we expand the attribution vectors by adding columns with values sampled from a uniform distribution between 0 and 0.5, which mimics the behavior of noises. We add the number of noise columns ranging from 1000 to 10000 to examine whether a K Means clustering can group the attributions into two groups the same as the assignment in Figure 2. We run K means clustering with and without PCA multiple times and record the accuracy in terms of Rand index [77] as well as the average run time. The result can be seen in Figure 3. The result shows that by transforming the attribution vectors with PCA, the clustering results become robust to the effect of sparsity among the attributions, which makes the subpopulation generation feasible from the local attribution data.

5 **SYSTEM DESIGN**

With the subpopulation generated from the local attributions, as discussed in Section 4.3, we present an interactive visualization system, **SUBPLEX**, with coordinated views to support the exploration of attribution groups. It consists of (a) a projection view that maps the attributions onto a 2D plane and (b) a subpopulation view that summarizes the attribution values from each cluster. These views act as the primary visual understanding channels of the explanations from the model and the
dataset (R.2). A categorical color scheme is used to encode each subpopulation throughout the whole system.

Fig. 5. Projection technique of local attribution: For each subpopulation, a fixed number of control points are extracted and they are mapped to the visual space using Multidimensional Scaling (MDS). Control points guide the projection of the remaining points using the Local Affine Multidimensional Projection (LAMP) technique.

5.1 Projection View
The projection view maps all attribution vectors in a two dimensional layout (Figure 4(A)). While projection techniques like Multidimensional Scaling (MDS) and t-SNE are popular choices, we have opted for a projection technique that best fits subpopulation analysis, the Local Affine Multidimensional Projection (LAMP) [34]. Since cluster labels are provided for each attribution vector,

Fig. 6. Comparison of the three dimensionality reduction techniques (LAMP, T-SNE, MDS) from the perspectives of visual representations and time performance: (A) shows the scatter plots based on the three projection techniques, where three clusters/groups can be clearly observed in all projection results; (B) illustrates the efficiency of three techniques in the line chart, where LAMP shows the best scalability in terms of time.

J. ACM, Vol. 37, No. 4, Article 111. Publication date: August 2018.
SUBPLEX: Towards a Better Understanding of Black Box Model Explanations at the Subpopulation Level

A supervised dimensionality reduction method can be employed to perform the mapping while preserving/emphasizing cluster structures [68]. The LAMP technique relies on a set of control points to map high-dimensional data to the visual space. More specifically, each control point has a weight associated with each point mapped by LAMP. The larger the weight, the closer to the corresponding control point the point is mapped. In order to further emphasize the clusters, weights between points and control points from the same class are increased (in our implementation weights are increased in 30%) while weights of outer-class control points are not changed. Control points are randomly chosen from each class and mapped by classical MDS [10], also shrinking inner-class distances in 30%. The procedure is illustrated in Figure 5. It can be clearly seen that with fewer control points, the projection creates a clear separation that allows cluster structure easier to be seen in the final projection layout. Furthermore, the medoid of each subpopulation (i.e., point with lowest pairwise distance within the group) is encoded as a clickable square so that when it is clicked, the points in the subpopulation will be highlighted.

Our motivation for using LAMP as the projection technique can also be illustrated in Figure 5. We generate a synthetic dataset with 3 clusters and 30 attributes and compare the speed and performances among LAMP, MDS, and tSNE. While all of the projection outputs are similar, we can see that LAMP has a much faster running time (R.1). Given an interactive workflow provided by the system, interactive computations are more desired.

Besides, identifying outliers is a vital operation when browsing a projection (R.3). To increase the stimulus of an outlier, we provide a function to highlight the outlier detected by outlier algorithms in the projection so that the projections can be more informative.

5.2 Subpopulation View

The subpopulation view provides detailed information for the properties of each group of the subpopulations (Figure 4(B)). The details are shown as a list of feature importances depicted with bar charts and histograms. The bar chart (Figure 4(B)(i)) shows the average attribution value of a feature among all points in a subpopulation group juxtaposed horizontally. While the histogram (Figure 4(B)(ii)) shows the distributions of the points in each subpopulation group in a superposed layout. Each distribution’s values (i.e., height) are normalized by the size of its subpopulation.

To facilitate the exploration of data in different priorities (R.4), sorting is provided for each of the columns. For the columns regarding each subpopulation, SUBPLEX sorts by the values (i.e. the length of the bar). However, to sort the distributions, we aim at prioritizing the distributions that deviate much across different subpopulations. To calculate the distances between two distributions, we use the earth mover’s distance (EMD) [75]. It briefly refers to the minimum amount of work to transform one distribution to another by moving the “distribution mass”. Given the distance metric, the distributions with a more significant sum of pairwise distances will be given a higher priority.

5.3 Interaction

Interaction plays an important role in facilitating data exploration between two views and human-in-the-loop analysis to provide synergy to the visual outcomes and results (R.5). The whole subpopulation analysis is an interactive computational workflow that a user can first define a number of partitions, then he can refine the final partition by brushing and filtering the instances in the system. SUBPLEX supports the following user interactions (Figure 7):

- **Brushing**: Brushing is enabled for users to select a subset of attribution vectors in the projection view. The system provides a lasso selection so that users can draw an irregular shape to include a group of potentially similar points (Figure 7(A)). To examine the behavior of selected
Fig. 7. Interacting with multiple coordinated views to refine the subpopulation result. (A) Selecting the subset of attribution view in the projection view. (B) Inspecting the details of the selected subset. (C) Adding or removing the subset as a new subpopulation.

- Adding and removing subpopulations: After inspecting the details such as the average attribution values and distributions for each feature for the selected subset (Figure 7(B)), users can extract the subset as a new subpopulation (Figure 7(C)) so that the subset now exists as an individual group in the system (i.e., have a new color, bars, and distributions).

5.4 Integration into Jupyter notebook
The visual analytics system is designed as an extension for data platform like Jupyter notebook, since we aim at creating a seamless workflow between model development and model understanding. The system provides the following API calls to extract the information in the visual analytics...
Fig. 8. System architecture as a widget integration into Jupyter notebook. Information such as ML model results are fed into the system and subpopulation output can be fed into notebook as variables.

System Integration in Jupyter

(A) Select attributions in the visual analytics system through function call

(ii) Get sub-population statistics of the selected instances

(B) Access the information of selected attributions as Python variables

Fig. 9. Apart from the visual analytics interface, SUBPLEX also provides APIs to transform information between the interface and Jupyter notebook. (A) Users can set the selection in the interface by passing a list of indices. (B) Users can output the information of the selected attributions in the interface as (i) a Pandas data frame containing all selected attributions as instances or (ii) the overall statistics of the selected attributions within each subpopulation.

platform or interact with the platform programmatically for a customized data inspection and analysis (Figure 9).

- `set_selection(data)`: As it might be infeasible or uncertain to select the attributions only through brushing and clicking, users can also select the attributions by passing an array of indices to this function to highlight the selection programmatically (Figure 9(A)).
- `get_selected_instances()`: When users select a subpopulation by clicking the medoid (i.e. square in the projection) or brushing a subset of attributions in the projection view, users
can call this function in the notebook to return the indices of the highlighted attributions as a Pandas dataframe (Figure 9(B)(i)).

- get_selected_groups(): Similar to the above function, users can call this function to return the aggregated subpopulation attribution values from the highlighted subset as a Pandas dataframe (Figure 9(B)(ii)).

5.5 Implementation
In this work, we implement a Jupyter Widget (ipywidget), using D3[11] and Backbone\(^1\) framework for visualization. Apart from supporting the projection results generated by LAMP[34] and clusters identified by K-means clustering[71] in default, we enable user-defined clustering labels and projection results to be visualized in this widget.

6 USE CASE SCENARIO
In this section, we demonstrate three usage scenarios regarding the use of SUBPLEX to address the interpretability goals of machine learning experts in understanding important features, investigating the bias features, and debugging the model (G.1-3). We used a credit score evaluation dataset [23] consisting of 6,600 applicants with 37 features and trained a neural network based on the application result (accept/ reject).

6.1 Use Case 1: Finding Important Features in Subpopulations
The first use case explores how our domain experts use SUBPLEX to identify the model’s behavior through different granularity of subpopulation explanations.

\(^1\)https://backbonejs.org/
Preparing subpopulations through interactive clustering and data cleaning. To begin with, our expert first imports the attributions to the system and tries to cluster them with a different number of clusters. Each clustering and projection process takes around three seconds in total. Then, he identifies that the original attribution data has five clusters with visibly distinct behavior in the detail view (Figure 10). Among the clusters, he discovers each has different sets of high-valued attributions, except one that has no significant attributions at all. Based on the definition of attribution, these instances are the ones that are hard to be explained by the explanation model (Figure 10(1)). Thus, as a data cleaning perspective, our expert selects the cluster by clicking on the medoid, then filters the instances in the data frame to remove them from the widget (Figure 10(2)).

Identifying significant features among subpopulations. After removing the instances with low attribution values, our expert discovers five unique rationales between the model and the dataset from the subpopulations. By sorting the attribution values for each subpopulation, he identifies each group’s characteristics by the long bars in the detail view (Figure 10(3)): the first group contains high attributions on the features related to the number of recent inquiries (“MSince-MostRecentInqexcl7days”) and delinquent trades (“NumTrades60Ever2DerogPubRec”); the second group contains features related to the customer’s age of trade lines (“AverageMInFile”); the third group consists of features regarding risk estimates (“ExternalRiskEstimate”); the fourth group is about the features concerning the absence of delinquency record (“MaxDelq2PublicRecLast12M = ‘unknown delinquency’”), and; the final group is concerned about the existence of delinquency (“MaxDelq2PublicRecLast12M=’30 days delinquent’”). The results reveal that while the model has a diversified rationale on different portions of the dataset, each rationale contributes to some unique traits to evaluate the credit risk with different perspectives.

Exporting and preparing the results. As a result, the expert exports the result to the data frames by clicking the medoids. While the visual exploration is completed, he proceeds to refine the final visual results by plotting the instances with static visualization libraries like matplotlib. The static charts are then shown in other presentation formats like PowerPoint for communications in future internal meetings. All in all, SUBPLEX provides a comprehensive visual exploration of the model’s attributions while being used tightly in the same programming environment.
6.2 Use Case 2: Evaluating Model Performance After the Removal of Bias Features

The second use case is concerned about how our domain expert pushes the model to explore new rationale by removing the useful features identified in the previous experiments.

**Removing the Useful Features.** To remove the useful features, the expert selects the above features to replace the values with random numbers. Therefore, when training the model using this dataset, the attributions of these features become negligible. To explore the outcome of this model, our expert imports the attribution data to the system to explore different groups of attributions.

**Evaluate Model’s Capability on Different Subpopulations.** After some experimenting, our expert discovers a clear separation of instances in the projection view when the number of clusters is set to two. The characteristics of the two clusters are very obvious in the detail view. One cluster has two strong feature attributions that are related to the absence of delinquency ("MaxDelq2PublicRecLast12M='current and never delinquent'" and “MaxDelqEver='current and never delinquent'”). Another one has no significant features at all. Therefore, by exporting the subpopulations and inspecting the cluster sizes, our expert understands that the model does not make consistent decisions on two-third of the dataset. For the remaining ones, it uses the clean delinquency record as the basis to make decisions. Thus, our experts summarize the influences of the important features in the dataset as the rationale for the customers without a clean delinquency record. He also saves these two different populations in separate files for further experiments.

6.3 Use Case 3: Debugging Model’s Architecture through the Adding Noisy Features

The last case focuses on the aspect of model debugging, of which the domain expert attempts to influence the model by including noisy and meaningless features adversely. It is done by adding features with values sampled from a normal distribution to the dataset.

**Inspecting Model’s Attributions.** After training the model and generating the attributions, our experts inspect the subpopulations of the attributions in the detail view (Figure 12). By sorting features in each cluster, our experts identify an interesting observation. While the clusters with clear rationale (i.e., long bars in some features) do not have high values among the noisy features, the clusters without clear rationale seem to have relatively longer bars on these noisy features.
expert then groups all the similar instances throughout different clusters to obtain a finer view (Figure 12(2)).

**Insights and Actions from the Observations.** Thus, our expert obtains the following insights: for the instances that do not follow mainstream rationale, they seem to be easier to be affected by noisy features. From a neural network perspective, this makes many senses. The unique behavior does not affect the gradients inside the network during batch processing due to its small population size. As a result, our expert decides to explore the possibility of data augmentation to generate more similar data to increase the adaptation of general logics of these highly customized instances. Moreover, he also reports the findings to caution the use of model when making niche decisions.

7 USER EVALUATION

To better understand how SUBPLEX is applied to ML model interpretation in general, we conducted semi-structured interviews with additional data scientists. The interview consisted of a go-through and open-ended discussion for every visual component and interaction of the system, and aimed at addressing the following usability questions:

- **Q1** How do general data scientists perceive the tasks (T.1-3) by subpopulation analysis?
- **Q2** How do data scientists perceive each visual component in terms of model interpretation?
- **Q3** What do data scientists prefer for visual analytics on model interpretation?

7.1 Participants

We interviewed 5 data scientists (two male, three female). The participants had experience building models ranging from three months to five years. In the following sections and paragraphs, we will use the title “scientist” to refer to any interviewee, since their jobs primarily focused on ML model development. Our recruitment goal, to avoid sample bias, is to seek a diverse pool of candidates to provide general impressions of subpopulation analysis in model interpretation but not to quantify any task effectiveness from the general public. To convey the results in statistics and numbers, other methods, such as quantitative usability tasks and surveys, could complement our findings.

7.2 Interview Design

The interview duration was one hour long per participant. Each participant first received an introduction of the system and the dataset (i.e., the credit scoring system used in Section 6) used in the demonstration. Once the users are familiar with the settings, we let them explore the system and dataset and explain to the interviewer the functionalities of different components in the interface. They were asked the impressions and concerns of the interface and suggested the usefulness and relevance to model interpretability.

8 RESULTS

8.1 Usefulness on Solving Three Use Cases

**Idea generation from subpopulation comparisons.** When the participants used SUBPLEX to explore the attribution subpopulations, they constantly compare different features among different subpopulations to identify whether some features are prevalent after bias removal or adversarial attacks. They observed some surprisingly high attribution values in the features that the developers permuted in one or two subpopulations. Thus, they raised concerns that the ML models were overfitted, data leakage problems happened, and the explanation method did not generate legitimate explanations. It provides us some insights into hypothesis generation enabled by such tools and workflow. While the process of interpretation is not standard, we recognize the process of generating explanations
as a creative process that involves lots of judgments, questions, and suggestions in which the interpretation methods will also be judged. Instead of giving an explanation to describe the behavior of each instance, providing multiple explanations at a time increases the concerns on the performance of the workflow and models, which corroborates with existing work [18, 29, 42] that there is a need to increase users’ considerations while developing insights from the models.

We also observed an additional consideration of granularity when evaluating the model explanations. Participants often selected outliers in the projection view to inspect the distributions of points that were not close to the center. They were used to understand the model performances by observing the behavior of the majority of the data and derived reasoning from groups of similar points. With projections provided, they were more eager and curious to select a subset of corner points and questioned on those points’ features. These provide us the insight that by applying subpopulation analysis, anomaly data in the visual interface will receive more attention. Also, participants mentioned that the tool provided them with the idea of population segmentation when browsing different subpopulations.

8.2 Perception of Visual Components for Model Interpretability

Pursuit of simplicity on system interaction. Our participants had undergone lots of trial and error processes during the exploration of SUBPLEX’s functionality. They first tried to understand the projection by selecting different subsets of points through brushing, then they output the subsets and inspect the statistics carefully to see if the different results provided some distinctions among the data. Some of the participants mentioned that although the interface was simple and intuitive, they need to have extra efforts to correlate the visual cues with the details of model explanations to summarize the behavior of ML models on this dataset. As a result, we observe that simplicity helps to remove the burden of visual understanding so that users can have more bandwidth to focus on model interpretation.

Trade-off between trust and efficiency on visual encodings. During the exploration, the visual component that all participants paid great attention to was the projection view. Most of them expressed skepticism towards the spatial layout because they had knowledge of dimensionality reduction techniques. However, they all agreed that it was troublesome to inspect all features in the subpopulations because it was difficult to remember and analyze many features at once. For example, one participant mentioned, “...The most confusing thing is again what are the points... the location of these points... like what does this space actually means... it seems quite abstract to me right now.” The projection has been related to concerns about trust [79] and this has to be carefully handled in the case of interpretation. As such, techniques have been prevalent among many clustering and dimensionality reduction tasks in visual analytics. This response motivates further studies to evaluate human trust in combining explanation and clustering processes.

Flexibility between programmable interface and visual analytics interface. Our participants questioned the methodology behind the subpopulation generation when they were exploring the attribution data. Also, they paid a considerable amount of attention to the distribution plots to explore further details of the features in the subpopulations. As a result, they required the statistics to be output to compute more details that they used in their daily operations. The feedback of participants suggested the importance of integrating a visual analytics system inside the loop of the programming platform. The trust of interpretation models could be improved if users are granted more engagement to the data exploration pipeline. One participant mentioned, “...the ranking is interesting. I do not trust is because I do not how the numbers are generated. Maybe I can export the distributions to see how values are generated... say, shapely value, or min/max value, Partial Dependency Plots...”
8.3 Visual Analytics for Model Interpretation

*Relationship between visualization literacy and ML model interpretability.* Some of our participants had raised concerns about the encodings of the projection view. The first question they asked was, “where are the axes in the scatterplot?” And after we explained that the points were projections of the data, they continued by questioning, “so what are the locations of the points mean?” After we explained that projections were 2D planes that approximated the similarities among the points, the participants showed great interest in such a visual data mining technique. One participant mentioned, “Maybe send me like a little bit more information about how dimensionality reduction is calculated. It is absolutely interesting.” Therefore, we observe that to address interpretability through visual analytics, users need to know how to interpret the visual encodings first. Although encoding numbers into visual encodings enables a more intuitive reasoning process, it is important to make sure the visualization is well taught towards the users first.

*Visual analytics mantra in model interpretation.* We observed our participants on the use of projection and detail table present the subpopulation information. Our participants often analyzed the data with the following steps: they first observed an overview of the whole dataset in the projection. Then they analyzed each subpopulation by switching the rankings according to the subpopulation being inspected. The model interpretation from such workflow helped establish model understanding similar to the visual analytics mantra [82]: “overview first, zoom and filter, then details on demand.” Our initial observation suggests that further explanation models could provide the data representation in such a way to achieve a well-rounded understanding across ML models and input data.

9 LESSONS LEARNED

We have learned two lessons in the process of collaborating with the machine learning experts. First, it is more and more important to integrate a visual analytics tool into a development environment where data scientists are familiar with and train their models. At the beginning of this project, we went through a few iterations of the visual system on the web, that is, building the tool as a traditional web application hosted on a local or remote server. However, our collaborators propose to make it an interactive jupyter widget because they want to stay in the environment of the jupyter notebook where they build the machine learning models. Data scientists are familiar with the coding workflow. So we need to enable them to interact with the visual analytics tool using a way they are used to. Another drawback of using an extra web application is that it requires additional I/O operations such as saving data to files and uploading the data to the server. However, staying in the development environment makes it much easier to transfer the data to be used for visualizations. Moreover, it is also flexible to get the desired data from the tool that the users can make more exploration later on. For example, it is convenient to get an array of data points that are selected by brushing in the projection view, which enables our users to do more analysis on the selected subset using native python functions.

Second, data scientists want to know the details of the necessary data processing steps when generating explanations. In our work, we use clustering and dimensionality reduction methods to assist the subpopulation analysis. During the iterations of the tool development, we are required to add processing information for processing steps. We first added the textual information about what the processing steps are (e.g., running clustering, running dimensionality reduction). After we conducted a few interviews with users, we realized that they also want to know the algorithms for clustering and dimensionality reduction we are using, as well as the parameters for each processing step. So in our latest version of the tool, we enable data scientists to initialize the widget using the customized objects of clustering or dimensionality reduction algorithm.
10 LIMITATION

Diversity of Tested Domains. In this work, we only worked with the FICO dataset. Although the visualization and interaction designs of our system is formed by multiple interviews and collaborations with experts and data scientists, it is essential to provide model interpretation to many other domains such as medicine and criminal justice. At present, our tool can be generalized to explain any user-defined tabular data from different domains. Increasing feedback from more application domains can help further development of our explanation tool. Meanwhile, the present tool only supports numerical data, which limits the usage of our approach in tasks such as image classification or speech recognition.

Lack of Quantitative Studies. Another limitation comes from the lack of quantitative studies. Although the interviews with experts are insightful, a well designed quantitative study can assist us to understand the merits and demerits more precisely. For instance, we can evaluate the performance of tasks, as proposed in section 3.1, from a more objective perspective.

Explanation of Projection. An intrinsic limitation of the dimensionality reduction results from the unfamiliarity to data scientists. On the one hand, the multidimensional projection (MDP) is a simple and straightforward way of presenting an overview of multidimensional data. On the other hand, some data scientists are not familiar with the MDP techniques so that they are confused with the scaling and distances in the projection at first glance. This also limits their interaction with the projection view.

11 CONCLUSION AND FUTURE WORK

In this work, through an iterative design process with expert machine learning researchers and practitioners, we identified a list of goals and tasks of explaining a machine learning model, designed and developed a novel visual analytics tool in the Jupyter notebook environment to assist the exploration of machine learning model explanations at a subpopulation level. We conducted semi-structured interviews with five data scientists. Our results show that data scientists have many reasons for interpretability and like interactive explanations. Although some of them are unfamiliar with interactive visual approaches, in the beginning, they give positive feedback when performing the analytic tasks after training. From our study, it is clear that there is an intense interest in explanatory interfaces for machine learning while there is a lack of such tools. As discussed in the previous section, we spot a few limitations in this work. We are particularly interested in further adapting our approaches to data and tasks in more domains and investigating more options for visual explanations for model users.

REFERENCES

[1] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y Lim, and Mohan Kankanhalli. 2018. Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In Proceedings of the 2018 CHI conference on human factors in computing systems. 1–18.
[2] Basak Alper, Nathalie Riche, Gonzalo Ramos, and Mary Czerwinski. 2011. Design study of LineSets, a novel set visualization technique. IEEE transactions on visualization and computer graphics 17, 12 (2011), 2259–2267.
[3] Jamal Alsakran, Yang Chen, Ye Zhao, Jing Yang, and Dongning Luo. 2011. STREAMIT: Dynamic visualization and interactive exploration of text streams. In 2011 IEEE Pacific Visualization Symposium. IEEE, 131–138.
[4] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. 2015. Modeltracker: Redesigning performance analysis tools for machine learning. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM, 337–346.
[5] Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. 2010. Examining multiple potential models in end-user interactive concept learning. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 1357–1360.
[6] John Annett. 2003. Hierarchical task analysis. Handbook of cognitive task design 2 (2003), 17–35.
[7] Enrico Bertini, Luigi Dell’Aquila, and Giuseppe Santucci. 2005. Springview: Cooperation of radviz and parallel coordinates for view optimization and clutter reduction. In *Coordinated and Multiple Views in Exploratory Visualization (CMV’05)*. IEEE, 22–29.

[8] Alsallakh Bilal, Amin Jourabloo, Yao Me, Xiaoming Liu, and Liu Ren. 2017. Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 152–162.

[9] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. ‘It’s Reducing a Human Being to a Percentage’: Perceptions of Justice in Algorithmic Decisions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 377.

[10] Ingwer Borg and Patrick Groenen. 2003. Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement* 40, 3 (2003), 277–280.

[11] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3 data-driven documents. *IEEE transactions on visualization and computer graphics* 17, 12 (2011), 2301–2309.

[12] Lauren Bradel, Chris North, and Leanna House. 2014. Multi-model semantic interaction for text analytics. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 163–172.

[13] Nan Cao, Jimeng Sun, Yu-Ru Lin, David Gotz, Shixia Liu, and Huamin Qu. 2010. Facetatlas: Multifaceted visualization for rich text corpora. *IEEE transactions on visualization and computer graphics* 16, 6 (2010), 1172–1181.

[14] Doina Caragea, Dianne Cook, and Vasant G Honavar. 2001. Gaining insights into support vector machine pattern classifiers using projection-based tour methods. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 251–256.

[15] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1721–1730.

[16] Motahhare Eslami, Sneha R Krishna Kumaran, Christian Sandvig, and Karrie Karahalios. 2018. Communicating algorithmic process in online behavioral advertising. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 432.

[17] Christopher Collins, Gerald Penn, and Sheelagh Carpendale. 2009. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1009–1016.

[18] Enrico Bertini, Luigi Dell’Aquila, and Giuseppe Santucci. 2005. Springview: Cooperation of radviz and parallel coordinates for view optimization and clutter reduction. In *Coordinated and Multiple Views in Exploratory Visualization (CMV’05)*. IEEE, 22–29.

[19] Alyssa Glass, Deborah L McGuinness, and Michael Wolverton. 2008. Toward establishing trust in adaptive agents. In *Proceedings of the 13th international conference on Intelligent user interfaces*. ACM, 227–236.

[20] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. 2018. This looks like that: Identifying and communicating class hierarchy? in *Proceedings of the 21th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 251–256.

[21] Alfred Inselberg. 1985. The plane with parallel coordinates. *The visual computer* 1, 2 (1985), 69–91.

[22] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1721–1730.

[23] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. 2018. This looks like that: Identifying and communicating class hierarchy? in *Proceedings of the 21th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 251–256.

[24] FICO. 2018. Explainable Machine Learning Challenge. https://community.fico.com/s/explainable-machine-learning-challenge.

[25] Rebecca Fiebrink, Perry R Cook, and Dan Trueman. 2011. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 147–156.

[26] Enrico Bertini, Luigi Dell’Aquila, and Giuseppe Santucci. 2005. Springview: Cooperation of radviz and parallel coordinates for view optimization and clutter reduction. In *Coordinated and Multiple Views in Exploratory Visualization (CMV’05)*. IEEE, 22–29.

[27] Enrico Bertini, Luigi Dell’Aquila, and Giuseppe Santucci. 2005. Springview: Cooperation of radviz and parallel coordinates for view optimization and clutter reduction. In *Coordinated and Multiple Views in Exploratory Visualization (CMV’05)*. IEEE, 22–29.

[28] Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. ‘It’s Reducing a Human Being to a Percentage’: Perceptions of Justice in Algorithmic Decisions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 377.

[29] Ingwer Borg and Patrick Groenen. 2003. Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement* 40, 3 (2003), 277–280.

[30] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3 data-driven documents. *IEEE transactions on visualization and computer graphics* 17, 12 (2011), 2301–2309.

[31] Enrico Bertini, Luigi Dell’Aquila, and Giuseppe Santucci. 2005. Springview: Cooperation of radviz and parallel coordinates for view optimization and clutter reduction. In *Coordinated and Multiple Views in Exploratory Visualization (CMV’05)*. IEEE, 22–29.
[33] Radu Jianu, Adrian Rusu, Yifan Hu, and Douglas Taggart. 2014. How to display group information on node-link diagrams: an evaluation. *IEEE Transactions on Visualization and Computer Graphics* 20, 11 (2014), 1530–1541.

[34] Paulo Joia, Danilo Coimbra, Jose A Cuminato, Fernando V Paulovich, and Luis G Nonato. 2011. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2563–2571.

[35] Enric Junqué de Fortuny, Marija Stankova, Julie Moeversoms, Bart Minnaert, Foster Provost, and David Martens. 2014. Corporate residence fraud detection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1650–1659.

[36] Minsuk Kahng, Pierre Y Andrews, Aditya Kalro, and Duen Horng Polo Chau. 2017. Activis: Visual exploration of industry-scale deep neural network models. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 88–97.

[37] Minsuk Kahng, Pierre Y Andrews, Aditya Kalro, and Duen Horng Polo Chau. 2017. A cti v is: Visual exploration of industry-scale deep neural network models. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 88–97.

[38] Eser Kandogan. 2000. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of the IEEE Information Visualization Symposium*, Vol. 650. Citeseer, 22.

[39] Daniel A Keim. 2002. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics* 8, 1 (2002), 1–8.

[40] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2017. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279* (2017).

[41] Josua Krause, Aritra Dasgupta, Jordan Swartz, Yindalon Aphinyanaphongs, and Enrico Bertini. 2017. A Workflow for Visual Diagnostics of Binary Classifiers using Instance-Level Explanations. *Visual Analytics Science and Technology (VAST), IEEE Conference on* (Oct 2017).

[42] Josua Krause, Adam Perer, and Enrico Bertini. 2018. A user study on the effect of aggregating explanations for interpreting machine learning models. In *ACM KDD Workshop on Interactive Data Exploration and Analytics*.

[43] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 5686–5697.

[44] Bum Chul Kwon, Ben Eysenbach, Janu Verma, Kenney Ng, Christopher De Filippi, Walter F Stewart, and Adam Perer. 2018. Clustervision: Visual supervision of unsupervised clustering. *IEEE transactions on visualization and computer graphics* 24, 1 (2018), 142–151.

[45] Isaac Lage, Andrew Ross, Samuel J Gershman, Been Kim, and Finale Doshi-Velez. 2018. Human-in-the-loop interpretability prior. In *Advances in Neural Information Processing Systems*. 10159–10168.

[46] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1675–1684.

[47] Antonio Lavecchia. 2015. Machine-learning approaches in drug discovery and data mining. *Information visualization and visual data mining*. ACM, 1675–1684.

[48] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2018. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[49] Q Vera Liao, Daniel Gruen, and Sarah Miller. 2020. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. *arXiv preprint arXiv:2001.02478* (2020).

[50] Brian Y Lim, Anind K Dey, and Daniel Avrahami. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2119–2128.

[51] Zachary C Lipton. 2018. The mythos of model interpretability. *Queue* 16, 3 (2018), 31–57.

[52] Bengmchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, and Shixia Liu. 2016. Towards better analysis of deep convolutional neural networks. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 91–100.

[53] Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. 2017. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics* 1, 1 (2017), 48–56.

[54] Shixia Liu, Jiannan Xiao, Junlin Liu, Xiting Wang, Jing Wu, and Jun Zhu. 2017. Visual diagnosis of tree boosting methods. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 163–173.

[55] Gang Luo. 2016. Automatically explaining machine learning prediction results: a demonstration on type 2 diabetes risk prediction. *Health information science and systems* 4, 1 (2016), 2.
Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

Gladys MH Mamani, Francisco M Fatore, Luis Gustavo Nonato, and Fernando Vieira Paulovich. 2013. User-driven feature space transformation. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 291–299.

Yamamoto Masafumi, Yoshikawa Tomohiro, and Furushashi Takeshi. 2010. Study on effect of MOGA with interactive island model using visualization. In *IEEE Congress on Evolutionary Computation*. IEEE, 1–6.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 63, 2 (1956), 81.

Yao Ming, Huamin Qu, and Enrico Bertini. 2018. RuleMatrix: visualizing and understanding classifiers with rules. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 342–352.

Yao Ming, Panpan Xu, Furuui Cheng, Huamin Qu, and Liu Ren. 2019. ProtoSteer: Steering Deep Sequence Model with Prototypes. *IEEE Transactions on Visualization and Computer Graphics (2019)*.

Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. 2019. Interpretable and Steerable Sequence Learning via Prototypes. (2019).

Ceena Modarres, Mark Ibrahim, Melissa Louie, and John Paisley. 2018. Towards Explainable Deep Learning for Credit Lending: A Case Study. *arXiv preprint arXiv:1811.06471* (2018).

Christoph Molnar. 2019. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/.

Luis Gustavo Nonato and Michael Aupetit. 2018. Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE transactions on visualization and computer graphics* (2018).

L. G. Nonato and M. Aupetit. 2019. Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Organization. *IEEE Transactions on Visualization and Computer Graphics* 25, 8 (2019), 2650–2673.

Eli Packer, Peter Bak, Mikko Nikkilä, Valentin Polishchuk, and Harold J Ship. 2013. Visual analytics for spatial clustering: Using a heuristic approach for guided exploration. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2179–2188.

Devi Parikh and C Zitnick. 2011. Human-debugging of machines. *NIPS WCSSWC* 2, 7 (2011), 2.

Hae-Sang Park and Chi-Hyuck Jun. 2009. A simple and fast algorithm for K-medoids clustering. *Expert systems with applications* 36, 2 (2009), 3336–3341.

Emilee Rader, Kelley Cotter, and Janghee Cho. 2018. Explanations as mechanisms for supporting algorithmic transparency. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 103.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “ Why should i trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

Bertrand Rouet-Leduc, Claudia Hulbert, Nicholas Lubbers, Kipton Barros, Colin J Humphreys, and Paul A Johnson. 2017. Machine learning predicts laboratory earthquakes. *Geophysical Research Letters* 44, 18 (2017), 9276–9282.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 59–66.

Dominik Sacha, Matthias Kraus, Jürgen Bernard, Michael Behrisch, Tobias Schreck, Yuki Asano, and Daniel A Keim. 2018. Somflow: Guided exploratory cluster analysis with self-organizing maps and analytic provenance. *IEEE transactions on visualization and computer graphics* 24, 1 (2018), 120–130.

Jorge M Santos and Mark Embrechts. 2009. On the use of the adjusted rand index as a metric for evaluating supervised classification. In *International conference on artificial neural networks*. Springer, 175–184.

Hans-Jorg Schulz. 2011. TREEVIS. net: A tree visualization reference. *IEEE Computer Graphics and Applications* 31, 6 (2011), 11–15.

M Sedlmair, Matt Brehmer, S Ingram, and T Munzner. 2012. Dimensionality reduction in the wild: Gaps and guidance. *Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada, Tech. Rep. TR-2012-03* (2012).

Jessica Zeitz Self, Radha Krishnan Vinayagam, JT Fry, and Chris North. 2016. Bridging the gap between user intention and model parameters for human-in-the-loop data analytics. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. ACM, 3.

Jinwook Seo and Ben Shneiderman. 2002. Interactively exploring hierarchical clustering results [gene identification]. *Computer* 35, 7 (2002), 80–86.

Ben Shneiderman. 1996. The eyes have it: A task by data taxonomy for information visualizations. In *Proceedings 1996 IEEE symposium on visual languages*. IEEE, 336–343.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org.
[84] Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. 2009. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies* 67, 8 (2009), 639–662.

[85] Hastie Trevor, Tibshirani Robert, and Friedman JH. 2009. *The elements of statistical learning: data mining, inference, and prediction*.

[86] Gaurav Trivedi, Phuong Pham, Wendy Chapman, Rebecca Hwa, Janyce Wiebe, and Harry Hochheiser. 2017. An interactive tool for natural language processing on clinical text. arXiv preprint arXiv:1707.01890 (2017).

[87] Edward R Tufte. 2001. *The visual display of quantitative information*. Vol. 2. Graphics press Cheshire, CT.

[88] F-Y Tzeng and K-L Ma. 2005. Opening the black box-data driven visualization of neural networks. In *VIS 05. IEEE Visualization*, 2005. IEEE, 383–390.

[89] Berk Ustun and Cynthia Rudin. 2017. Optimized risk scores. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1125–1134.

[90] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. 2019. Designing theory-driven user-centric explainable AI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–15.

[91] John Wenskovitch, Ian Crandell, Naren Ramakrishnan, Leanna House, and Chris North. 2017. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 131–141.

[92] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.

[93] Kanit Wongsuphasawat, Daniel Smilkov, James Wexler, Jimbo Wilson, Dandelion Mane, Doug Fritz, Dilip Krishnan, Fernanda B Viégas, and Martin Wattenberg. 2017. Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 1–12.

[94] Xiaoru Yuan, Peihong Guo, He Xiao, Hong Zhou, and Huamin Qu. 2009. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1001–1008.

[95] Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S Ebert. 2018. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 364–373.

[96] Yixuan Zhang, Kartik Chanana, and Cody Dunne. 2018. IDMVis: Temporal event sequence visualization for type 1 diabetes treatment decision support. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 512–522.