Provectories: Embedding-based Analysis of Interaction Provenance Data

Conny Walchshofer, Andreas Hinterreiter, Kai Xu, Holger Stitz, and Marc Streit

Abstract—Understanding user behavior patterns and visual analysis strategies is a long-standing challenge. Existing approaches rely largely on time-consuming manual processes such as interviews and the analysis of observational data. While it is technically possible to capture a history of user interactions and application states, it remains difficult to extract and describe analysis strategies based on interaction provenance. In this paper, we propose a novel visual approach for meta-analysis of interaction provenance. We capture single and multiple user sessions as graphs of high-dimensional application states. Our meta-analysis is based on two different types of two-dimensional embeddings of these high-dimensional states: layouts based on (i) topology and (ii) attribute similarity. We applied these visualization approaches to synthetic and real user provenance data captured in two user studies. From our visualizations, we were able to extract patterns for data types and analytical reasoning strategies.

Index Terms—Visualization techniques, Information visualization, Visual analytics, Interaction Provenance, Sensemaking

1 Introduction

Understanding the analytical reasoning process of users who work with interactive tools, in general, and with visualization tools in particular, has been an active research topic. One way to gain more insights into how users work with such tools is to record *interaction provenance* data, which describes the lineage of data, system states, visualizations used, and user interactions. It is typically recorded in the form of *protocols*, such as audio/video recordings [3], usage logs [18], and user notebooks [52]. In the human-computer interaction (HCI) community, these protocols are analyzed in an attempt to better understand a user's behavior and intentions [31].

In recent years, the visualization community has recognized the potential of insights gained from capturing [5], [11], [34], visualizing [6], [48], and interpreting provenance [46] from user interactions with visualization tools. According to the distributed cognition approach by Hollan et al. [24], a close relationship exists between users' activities and their thought processes. Pohl et al. [41] argued that visualizations of interaction provenance data can be used to make sense of users' reasoning processes. However, there are few approaches that support effectively the *meta-analysis* of analytic provenance as defined by Ragan et al. [43].

The primary contribution of our work is *Provectories*, an approach that helps visualization researchers, designers, and developers to better understand the behavioral patterns and analytic strategies of users. As shown in Figure 1, we transform application states of the interaction provenance into feature vectors and visualize them using two different types of embeddings: (1) a *topology-driven* layout that aims to show patterns of states based on their connectivity and (2) an *attribute-driven* layout that visualizes states based on their similarity. These embeddings give rise to visual patterns that can be related to specific user actions. Thus, we introduce a new encoding that supports the analysis of a single user state, a sequence of states, and the identification of similarities between two or multiple states through interactive exploration. Further, *Provectories* can be applied to a broad spectrum of use cases and tools, ranging from single interactive visualizations to feature-rich tools such as Tableau and Microsoft Power BI.

As secondary contribution, we describe the visual patterns that we extracted from *Provectories* visualizations of synthetic user interactions and of real-world user interactions from two user studies with different visual analysis tools. We describe insights gained from single sessions and patterns resulting from the combination of multiple sessions. We discuss the relative strengths and weaknesses of both layouts used in the *Provectories* workflow to identify interaction patterns in these visualizations.

We structured the paper as follows. In Section 2 we discuss existing approaches to interaction provenance representation and analysis. In Section 3 we present application scenarios and introduce an illustrative example. In Section 4 we describe the *Provectories* workflow conceptually; implementation details are given in Section 5. In Section 6 we present the results of applying *Provectories* to synthetic and real-world interaction provenance data and discuss the advantages of two different layouts. We then summarize the limitations of our new visual analysis approach in Section 7. Section 8 concludes the paper.

2 Related Work

In this section, we describe how interaction provenance has been defined in the literature and discuss why visualization researchers might study interaction provenance. We then

- *Conny Walchshofer, Andreas Hinterreiter, and Marc Streit are with Johannes Kepler University Linz, Austria.*
E-mail: {conny.walchshofer, andreas.hinterreiter, marc.streit}@jku.at.
- *Kai Xu is with Middlesex University London, UK.*
E-mail: k.xu@mdx.ac.uk.
- *Holger Stitz and Marc Streit are with datavisyn GmbH, Austria.*
E-mail: holger.stitz@datavisyn.io.

Manuscript received November XX, 2020

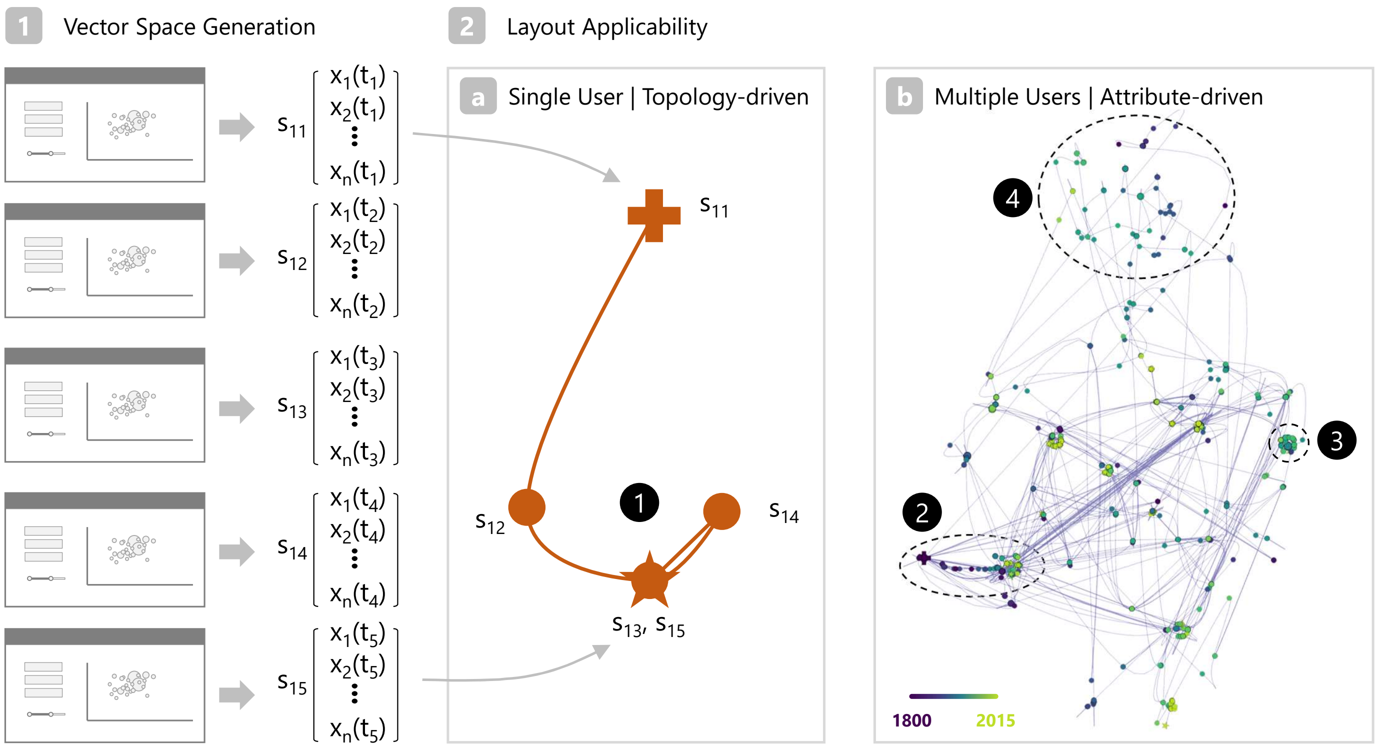

1
Vector Space Generation
2
Layout Applicability
s_{11}
$x_1(t_1)$ $x_2(t_1)$ ⋮ $x_n(t_1)$
s_{12}
$x_1(t_2)$ $x_2(t_2)$ ⋮ $x_n(t_2)$
s_{13}
$x_1(t_3)$ $x_2(t_3)$ ⋮ $x_n(t_3)$
s_{14}
$x_1(t_4)$ $x_2(t_4)$ ⋮ $x_n(t_4)$
s_{15}
$x_1(t_5)$ $x_2(t_5)$ ⋮ $x_n(t_5)$
a
Single User | Topology-driven
s_{11}
s_{12}
1
s_{14}
s_{13}, s_{15}
b
Multiple Users | Attribute-driven
4
3
2
1800
2015

Fig. 1: Identifying meta-analysis patterns for interaction provenance through 1 a schematic display of the vector space generation and 2 the influence of two layouts. a shows a single user session visualized with a topology-driven layout (force-directed) and b represents multiple user sessions visualized with t-SNE, an attribute-driven layout. The circular annotations highlight ❶ a loop back within the analysis process, ❷ a chain of numerical value changes, ❸ a selection of multiple countries, and ❹ sessions alternating the assignment of attributes on the x-axis, y-axis, and mark size. The colors of the states indicate numerical values from *1800* (●) in violet to *2015* (●) in light green.

discuss previous approaches to meta-analysis, in particular those based on visualizations of provenance data.

2.1 Interaction Provenance

Ragan et al. [43] introduced an organizational framework for different types of provenance in visualization and data analysis. They defined *interaction provenance* as "the history of user actions and commands with a system" [43, p.35]. There are various motivations for logging explicit and observable user interactions, such as selections, clicks, keystrokes, and mouse movement. Gotz and Zhou [19] introduce an action taxonomy consisting of three top-level categories that can be used to gain insights from provenance data, namely exploration, insight, and meta actions. Interaction logs can be used for the purposes of collaboration, reproducibility, storytelling, and retrieval [21], [47]. More closely related to our work, interaction provenance can be analyzed to understand how users interact with a visualization system [18] or to measure the effectiveness of a tool [9]. The process of making sense of such logs is referred to as provenance meta-analysis.

2.2 Provenance Meta-Analysis

Ragan et al. [43] described meta-analysis as one of six purposes for interaction provenance tracking. Xu et al. [53] provided a spectrum of possible reasons for conducting meta-analyses on provenance data. Reviewing an analysis process to understand the analytic strategies of users has been identified as an important task [11], [14], [43], which can be implemented in various ways.

Wei et al. [51], for instance, employed clickstream data to analyze purchase patterns. The data is labeled with predefined actions (e.g., selection of a category, setting a price) and analyzed based on the ordered sequence. Heer et al. [22] described how users interact with a visual analytics tool by evaluating aggregated collections of history sessions. Pohl et al. [32] qualitatively analyzed interaction provenance based on thinking-aloud protocols. They identified various strategies that users applied to interpret and understand visualizations: comparing, laddering, explaining (storytelling), summarizing, eliminating, and verifying. Similarly, Madanagopal et al. [30] analyzed interaction provenance from a sociotechnical perspective by conducting interviews. They elaborate on how analytical provenance can be captured and used by taking different end-users into account. However, they call for future work, as analytic provenance in contrast to data provenance is still in its infancy. With reVISit, Nobre et al. [35] analyzed interaction provenance by comparing event sequences using a node-link diagram and identified "multidrag", "sort and select", and "select and refine" as analysis strategies. *Provectories* aims to identify such user strategies as visual patterns. Thus, *Provectories* is a visualization-based approach to the meta-analysis of interaction provenance.

2.3 Provenance Visualization

According to a recent survey by Xu et al. [53], interaction provenance is most commonly encoded as a temporally ordered sequence. Visualizing interaction provenance in this way allows step-wise retracing of the individual interactions [5], [11], [14] and can thus convey the users' thought processes [29], [54]. However, sequential visualizations are less suitable for discovering patterns and relationships. They neither preserve interesting topological structures, such as loops or branches in a user's interaction path nor convey a potential similarity between application states visited. These issues are addressed by the *topology-based* and *attribute-driven* visualization techniques in *Provectories*.

2.3.1 Topology-driven Layouts

Provenance data can be treated as a graph, with nodes representing states of a data item or application and edges representing actions of users that lead to transitions between the states. Graph-based provenance visualization can reveal patterns such as branching, cycles, or commonly revisited states (i.e., nodes with high connectivity).

VisTrails [7] is a graph-based visualization of workflow provenance. *GraphTrails* [12] is an exploration tool for network analysis that incorporates interaction-provenance on the fly. *VizCept* [8] is a collaborative analysis system for textual data that allows users to keep track of each other's findings and relationships in a shared topological concept map. The *Knowledge-Transfer Graph* by Zhao et al. [55] shows a node-link visualization that aims to help researchers to externalize their thought processes in collaborative analyses. *reVISit* [35] assesses interaction provenance data based on both qualitative and quantitative data by showing interaction patterns and analysis strategies as event sequences.

Similarly, we use a force-directed graph layout for visually representing interaction provenance. In addition to this topology-driven layout, we also investigate and employ layouts in which the similarity between states determines the positions of the nodes.

2.3.2 Attribute-driven Layouts

The application states in a provenance log can be viewed as a high-dimensional time series rather than a graph. Bach et al. [2] proposed *TimeCurves* as a visualization technique for revealing similarity in high-dimensional time series. Time curves are trajectories through a two-dimensional embedding of the data points, which give rise to visual patterns such as clusters, cycles, U-turns and oscillations. Time curves are based on multidimensional scaling (MDS) for the embedding; similar visualizations can be constructed by means of other dimensionality-reduction techniques, such as PCA, t-SNE [50], and UMAP [33]. Time-curve-like visualizations have been used to visualize high-dimensional time series in a wide variety of application domains, for example, dynamic graphs by van den Elzen et al. [49] and neural networks by Rauber et al. [44].

In previous work [23], we used collections of time curves to visualize decision-making processes in games and puzzles, and described general patterns emerging in such visualizations. In this work, we use the same approach for visualizing interaction provenance in an attribute-driven layout. This makes our approach closely related to ModelSpace by Brown et al. [6]. ModelSpace is based on the concept of "numerical analytic provenance", which consists of sequences of vectors that describe the users' interactions with a system "via the proxy of changes to their underlying machine learning models." The authors also mention a possible application of ModelSpace to visual analytics systems in which the users do not interact with such models. However, from the brief discussion of the example application—a search interface for the Finding Waldo puzzle [5]—it is not clear how the feature space in such cases relates to the insights gained from the ModelSpace visualization. By applying our similar approach to two visual analytics tools with fundamentally different choices for the state representation, we aim to strengthen this connection. Furthermore, we show that additional visual encoding options and interaction techniques, such as tailored single-state and summary visualizations or a step-wise path analysis, can facilitate the interpretation of the projected provenance data.

3 Requirements and Usage Scenario

We designed *Provectories* for the purpose of understanding user behavior patterns and analysis strategies from interaction provenance. Gleicher [17] enumerated three ways of comparing sessions: comparison between two items, between "a few" items, and between many items at the same time. With *Provectories*, we aim to cover all three aspects, performing meta-analysis to understand (i) a single user's analytical process, (ii) similar analysis processes by single or multiple users, and (iii) similar approaches by and between multiple users. Thus, *Provectories* uses two layouts to enable comparison between unique states from a single session, between unique states from multiple user sessions, and between contiguous states from multiple user sessions.

Single-session investigation focuses on understanding behavioral patterns and the overall analysis strategy of a single user. This type of investigation aims to answer questions such as whether a user encountered difficulties during the analysis and whether the user had a systematic search strategy or performed a rather untargeted exploratory analysis.

Multi-session investigation builds on single-session investigation, but focuses on the comparison of the interaction provenance from multiple users working with the same tool. Here, the goal is to understand the similarities and differences in analysis behavior between the users. This type of investigation aims to answer questions such as whether many users encounter the same difficulties, or how effective different analysis strategies are. Multi-session investigation can be divided into comparing sessions in which users perform (i) the same or similar tasks or (ii) different tasks.

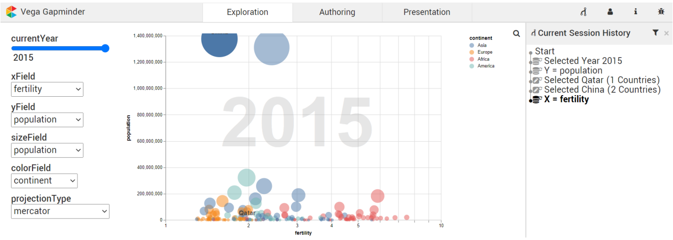
3.1 Requirements

We derive the following requirements for single and multiples sessions from the existing literature [27], [37] and our prior research experience [13], [23], [53]. To support single-session investigation, *Provectories* is designed to:

S1 show the entire analysis sequence from beginning to end, following the temporal order;

S2 include the user interaction and/or system state information, such as the changes between two consecutive steps in the analysis sequence;

S3 facilitate the analysis of data coverage during the exploratory analysis, such as the data trails that lead the user from the starting point to the final answer.



currentYear: 2015

xField: fertility

yField: population

sizeField: population

colorField: continent

projectionType: mercator

Current Session History
- Start
- Selected Year 2015
- Y = population
- Selected Qatar (1 Countries)
- Selected China (2 Countries)
- **X = fertility**

Fig. 2: Interface of the *Gapminder* visual analytics tool [47] with the history graph.

Further, we target to identify whether a user focuses on certain data attributes and/or part of the dataset or more widely explores the entire data space;

S4 facilitate the investigation of any analysis tactics or strategies that the user deployed, such as whether the user explores the data space randomly or follows a certain strategy. This also includes the identification of situations such as a user getting stuck at a certain stage of the analysis, which could be indicated by revisiting certain visualization states.

To support multi-session investigation, *Provectories* aims to:

M1 provide an overview of all the analysis sessions, such as which part of the dataset is more frequently investigated and where does most of the unsuccessful analyses ended up;

M2 support the comparison among analysis sessions, e.g., do successful analysis sessions share similar exploration pathways and if there is any common difference between successful and unsuccessful sessions;

M3 facilitate the discovery of other sense-making patterns, such as whether more efficient analysis sessions can be identified by certain visual patterns and is there any correlation between the investigation strategy and data attributes/subspace.

3.2 Usage Scenario

We hereafter use *Gapminder* [45], [47] as a guiding example to explain how *Provectories* works. The Gapminder tool allows users to explore the development of countries over time. As outlined in Figure 2, it consists of a bubble chart in which each country is represented by a colored mark. Users can interactively map attributes, such as GDP, life expectancy, and child mortality, to either one of the axes or the size of the country marks, and change the year between 1800 and 2015 with a time slider. At any time, the application state can be fully described by the following information: the timestamp of the interaction; the data attributes mapped to x-axis, y-axis, mark size, and mark color; the year selected (between 1800 and 2015); and any countries selected.

In a simple analysis of the relationship between population and fertility among countries in 2015, the user can perform the following steps: (1) change the year to 2015; (2) change the data attribute for the y-axis to *population*; (3) select the country *Qatar*; (4) add *China* to the country selection; and (5) change the data attribute for the x-axis to *fertility*. This analysis results in the five applications states listed in Table 1.

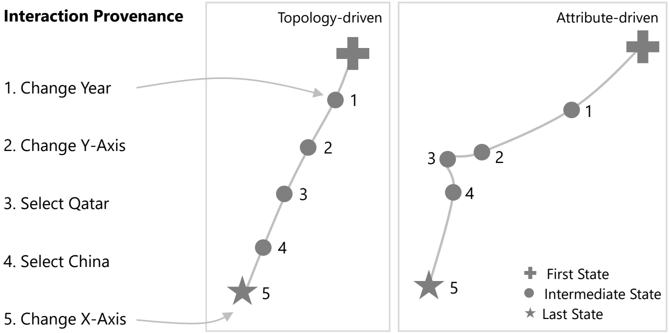


Fig. 3: Schematic illustration of the Gapminder usage scenario explaining how application states are mapped in the two layout variants.

For the purpose of meta-analysis, we use the sequence of application states visited by a user and display the interaction provenance in two layouts, see Figure 3. In both representations, ✚ and ★ indicate the beginning and the end of a session, respectively. When applying a force-directed layout (topology-driven), a chain of five successive states is visible. In contrast, the attribute-driven layout (calculated using t-SNE) places the states corresponding to selections of the countries (states 3 and 4) closer to each other than the other selected states. This is the result of an underlying "conceptual" or "analytical distance", which was defined to be smaller between states 2 to 4 than between the other ones.

4 Provectories

The fundamental workflow underlying the *Provectories* approach consists of three steps, as illustrated in Figure 4: 1 the application states resulting from one user's or multiple users' interactions with a visual analytics tool are recorded; 2 the application states are transformed to high-dimensional feature vectors; and 3 for the purpose of meta-analysis, the recorded analysis sessions are interactively visualized as trajectories through a two-dimensional embedding space based on various layout techniques.

4.1 Logging of Application States

As indicated in Figure 4 1, the first step in the *Provectories* workflow consists of creating user interaction logs for a given visual analytics tool. Each user interaction (of a predefined set of interactions) triggers the logging of the updated application state. The complexity of the visual analytics tool

TABLE 1
Interaction provenance for the *Gapminder* example. Here, x_0, y_0, s_0, and c_0 represent the default data attributes mapped to the axes, size, and color, respectively; Y_0 and C_0 represent the default initial selections for year and countries. **Bold text** indicates changes in the application state resulting from a user interaction.

Time	x-axis	y-axis	Size	Color	Year	Countries
t_0	x_0	y_0	s_0	c_0	Y_0	C_0
t_1	x_0	y_0	s_0	c_0	**2015**	C_0
t_2	x_0	**population**	s_0	c_0	2015	C_0
t_3	x_0	population	s_0	c_0	2015	**Qatar**
t_4	x_0	population	s_0	c_0	2015	Qatar, **China**
t_5	**fertility**	population	s_0	c_0	2015	Qatar, China

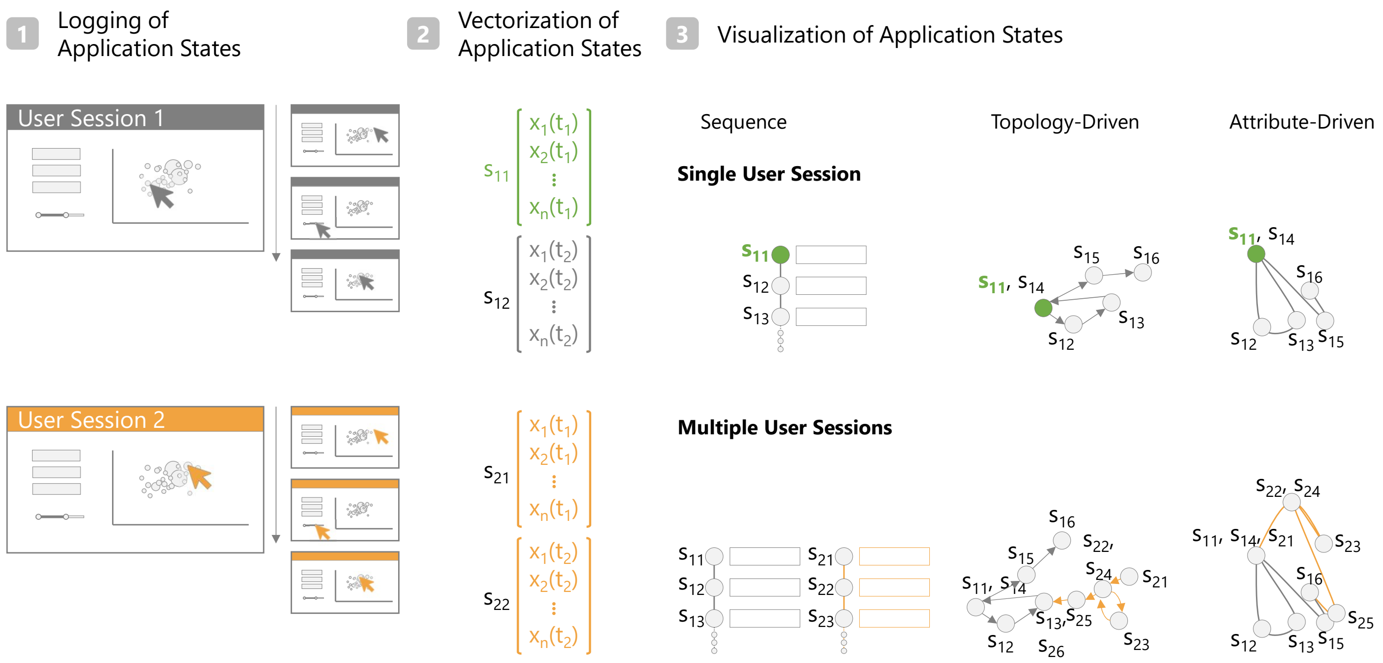


Fig. 4: *Provectories* workflow. (1) Interaction provenance is captured from a visual analytics tool. (2) Each interaction leads to a system state which is encoded as a feature vector such as s_{11}. (3) The sequence of provenance vectors is then visualized with a topology-driven or attribute-driven layout. Such a visualization can help to analyze a single session or to compare multiple sessions (such as Sessions 1 ■ and 2 🟧) concurrently.

and the goals of the subsequent meta-analysis determine the granularity of the application state and which interactions are to be logged. For the subsequent steps in the *Provectories* workflow, it is important that each user session is stored as a temporally ordered list (**S1**) of potentially unstructured or heterogeneous data items which can be transformed to feature vectors.

4.2 Vectorization of Application States

In the second step of the *Provectories* workflow, the logged application states are transformed into numerical feature vectors (see Figure 4 2). This transformation serves two purposes: First, it provides a structured way to determine equivalent states for the subsequent topology-driven layout (see Section 4.3.1). Instead of performing the similarity check directly on the complete and potentially complex logged states, this "quantization" introduces an optional abstraction and/or simplification step. Second, it enables the calculation of distances between application states (see Section 4.3.2).

We hypothesize that there are three fundamentally different ways of vectorizing application states:

1) **Vectorization via proxy**—In some applications, users' interactions with a visual analytics system transform an underlying object that may be readily represented as a vector. The changes to this object can serve as a proxy to understanding the user's actions. This type of vectorization is showcased in ModelSpace by Brown et al. [6]. If the underlying object is by itself not easily interpretable, as is the case with black-box machine learning models, this approach may introduce an additional layer of complication.
2) **Direct translation of interactive components**—In many visual analytics applications, each user interaction changes a variable of a given datatype. For instance, radio buttons relate to categorical variables, sliders to numerical variables, and the result of panning in a 2D plot may be represented by a 2D vector. In such cases, each variable of interest can be directly encoded and used as part of a compound vector representation that identifies the current application state.
3) **Hand-crafted, semantic representations**—In more complex cases, the effects of user interactions have to be translated to a numerical form by introducing a representation that conserves the semantics of the data that the user interacts with. The exact form of this representation depends strongly on the specific user interface, the actions tracked, and the tasks performed by the users.

The Gapminder usage scenario outlined in Section 3.2 lends itself well to using a compound representation constructed from direct translations of the interactive components. Here, the individual encodings depend on the data type:

Cat For categorical attributes, a simple one-hot encoding is the obvious choice.

Bool Boolean attributes can be treated as categorical attributes with two options or can be represented by a single integer that is either 0 or 1.

Num Numerical attributes require no further encoding.

Set Set attributes can be encoded by a sequence of zeros and ones of length l, where l is the cardinality of the complete set.

These encodings for categorical, Boolean, numerical, and set attributes introduced above give rise to simple distance metrics, as specified in Section 4.3.2. In many cases, however, the user interactions are too complicated to be directly translated, or the tracked information is too low-level to carry any information about the user's mental model. In such cases, it is necessary to derive higher-level features with better semantics. In Section 6.3, we describe such a tailored vector representation for the case of users interacting with a scatterplot through brushing.

4.3 Visualization of Application States

The final step of the *Provectories* workflow is the visualization of the paths that users take through the application-state space (see Figure 4 3). The high-dimensional feature vectors are embedded in two dimensions and visualized as a scatterplot using either a *topology-driven layout*, which emphasizes connectivity between identical states, or an *attribute-driven layout*, which focuses on the similarity between states. In both cases, each point represents an application state. The paths of users are visualized as trajectories going through these points. Data attributes or metadata can be mapped to the visual channels of the line and point marks. We first describe the embedding techniques and the required pre-processing of the provenance vectors. We then discuss the visual encoding choices and how meta-analysts can interact with the *Provectories* visualization.

4.3.1 Topology-driven Layout

For the topology-driven layout, we treat the collection of user sessions as a graph. We first determine a set of unique nodes, where each node represents a unique application state. Uniqueness is based on the identity of the high-dimensional feature vectors. We treat two nodes as connected if one succeeds the other in any of the user sessions. We lay out the nodes using a force-directed network spatialization algorithm (ForceAtlas2 [25]; for implementation details, see Section 5). The nodes are then connected by drawing a Catmull–Rom spline trajectory through them for each session (**M1**; see Section 4.3.3).

In the topology-driven layout, a single user session with no duplicate states always results in a linear "chain" of points. Only when states are revisited or shared across multiple sessions, patterns, such as loops and branches, emerge from this layout (**S3**, **M2**).

4.3.2 Attribute-driven Layout

For the attribute-driven layout, we treat the whole collection of application-state vectors across all user sessions as samples from a high-dimensional manifold. We embed these samples based on their similarity (**S3**, **M2**, **M3**), using various dimensionality reduction techniques. Specifically, we compare the results for MDS, t-SNE, and UMAP.

As these dimensionality reduction techniques aim to place similar points close to each other, it is important to define a meaningful metric for calculating the mutual distances between the high-dimensional feature vectors. For compound representations based on simple translations of interactive components, we suggest defining this distance metric based on individual distance functions for each attribute type. To this end, let α_i and β_i be the vectorizations of a single attribute with type $k(i) \in \{\text{cat}, \text{bool}, \text{num}, \text{set}\}$ for two different states A and B, as described in Section 4.2.

Cat For one-hot encoded categorical attributes, a reasonable distance metric can be defined via the inner product: $d_{\text{cat}}(A, B) = 1 - \alpha_{\text{cat}} \cdot \beta_{\text{cat}}$.

Bool For Boolean attributes encoded with a single number (0 or 1), the result of an exclusive or (XOR) can be used: $d_{\text{bool}}(A, B) = \alpha_{\text{bool}} \oplus \beta_{\text{bool}} = \alpha_{\text{bool}} + \beta_{\text{bool}} \pmod 2$.

Num For one-dimensional numerical attributes in compound representations, it makes sense to define the distance as the absolute difference normalized by the total value range $(n_{\max} - n_{\min})$ of the attribute: $d_{\text{num}}(A, B) = |\alpha_{\text{num}} - \beta_{\text{num}}|/(n_{\max} - n_{\min})$.

Set Set attributes may be compared using the Jaccard index. Alternatively, if the encoding described in Section 4.2 is used, a normalized p-norm of the difference between two vectors can be used as a distance function: $d_{\text{set}}(A, B) = \|\alpha_{\text{set}} - \beta_{\text{set}}\|_p / s^{1/p}$, where s is the cardinality of the complete set. See Section 6.1 for a short example and discussion of this choice.

Total Finally, the total distance between two states A and B in a compound representation can be calculated as the weighted sum of all individual attribute distances:

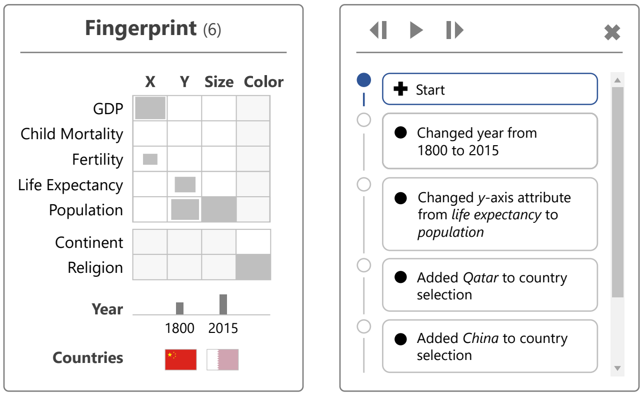
$$d(A, B) = \sum_i w_i \times d_{k(i)}(A, B). \tag{1}$$

By default, all weights are equal, i.e., $w_i = 1$, but they can be chosen freely by the meta-analyst depending on the scenario. The higher the weight of a specific attribute, the greater the likelihood that patterns for the associated data type will prevail in the embedding. This compound metric is similar to a weighted version of Gower's (dis)similarity [20].

In the case of a hand-crafted vectorization, the distance function must be chosen/constructed in such a way that the desired semantics are preserved (see Section 6.3 for an example). We use the pairwise distances as an input to the MDS, t-SNE, and UMAP techniques (Section 6.1 and 6.2). Unlike in our previous work [23], we remove duplicate high-dimensional vectors prior to the embedding by default. Otherwise, clusters of identical points can be mistaken for specific data-related patterns. In the case of compound representations, this removal of duplicates takes into account whether any of the weights are set to zero by the user. Zero-weighted attributes are treated as duplicates regardless of their value. We set the perplexity hyperparameter of t-SNE to 50 by default [50], and choose the nearest neighbor parameter for UMAP accordingly (since perplexity can be understood as a smooth measure of the number of nearest neighbors). Details of the implementations used for the embedding are given in Section 5.

4.3.3 Visual Encoding and Interactivity

As stated above, each layout technique results in a scatterplot of embedded application states. We visualize the user sessions as spline trajectories through these points (**S1**, **M1**). We chose this design over more traditional graph-drawing techniques (e.g., tree layout) for three reasons: First, drawing an individual trajectory for each session automatically results in an effective multigraph visualization in which parallel edges are visible as such. Second, each user session has its own distinct path, whose visual channels can be used to encode additional data. Third, in cases in which the meta-analyst decides not to remove duplicates in the attribute-driven layout, the same drawing algorithm can be applied.



+ Start

● Changed year from 1800 to 2015

● Changed *y*-axis attribute from *life expectancy* to *population*

● Added *Qatar* to country selection

● Added *China* to country selection

(a) Summary visualization

(b) History Graph

Fig. 5: (a) *Provectories* summary visualization for multiple selected application states, encoding the relative frequency of attribute values. (b) The history graph view for step-wise analysis of a single session.

Meta-analysts can select the visual encoding of the point and line marks. Point marks can be colored categorically depending on categorical or Boolean values, or using a sequential color scale for numerical values. An age attribute is also available which corresponds to the temporal index of each application state within its session. Lines can be colored categorically by meta-attributes such as usernames and predefined task labels. They can further be switched on and off by their categorical labels and filtered by length by using a range slider. These coloring and filtering options address *Explore Dimensions* and *Explore Items in Enriched Layout* tasks and described by Nonato and Aupetit [36].

To let meta-analysts inspect the underlying high-dimensional data for specific points, the *Provectories* visualization features so-called summary visualizations [13]. Upon hovering over a point, the summary visualization of the corresponding single application state is shown. When multiple points are selected (e.g., via a lasso selection), the summary visualization is adapted to encode the distribution of values among the application states. The exact visual encoding of the summary visualization depends on the number and types of attributes that describe the state of a given application.

Here, we describe the summary visualization designed for the Gapminder example (see Figure 5a). A table lists all categorical and Boolean attributes with their values, where the frequency of values within the selection is encoded by the size of the marker in each cell. A histogram shows the distribution of year values. The distribution of set selections is displayed as a list of country flags, with each flag's opacity encoding the number of states for which that country was part of the selection. This list is ordered by frequency, with the most frequently selected countries appearing first.

To facilitate tracing of individual user sessions even in the presence of potential visual clutter, the history graph with a list of user actions and the resulting application states for an individual session can be activated (see Figure 5b). In reVISit, Nobre et al. [35] refer to the sequential analysis of interaction states by a video-like experience using a playback feature. Selecting states in the history graph overlay the embedding space with arrows that indicate the position and direction of the user session. Analysts can follow the session in a step-wise manner or via an automated animation.

5 Implementation

The *Provectories* workflow is implemented as three individual components which closely correspond to the three steps of the workflow described in Figure 4: 1 a system for tracking the interaction provenance, which must be incorporated into the visual analytics tool that meta-analysts want to study; 2 a module that structures, processes, and exports the recorded provenance data; and 3 the interactive visualization of the user sessions.

For the first user study (Gapminder), we use the *KnowledgePearls* implementation of Gapminder [47] for provenance tracking. The resulting provenance files are processed in Python. We provide a Python module[1] with classes for application states, user sessions, and collections of sessions that can be adapted to interaction data from different visualizations. For the second user study (User Intent), we use the experiment data from Gadhave et al. [16] that was used by the authors to predict the users' intent based on the selection of data points in a scatterplot using the Trrack library [10]. We again provide a Python module[2]. The processed provenance data can be exported with or without pre-calculated embedding coordinates. We use the openTSNE implementation of *t*-SNE [42], the official UMAP Python implementation [33], the scikit-learn MDS implementation [38], and the ForceAtlas2 implementation from the datashader module [1].

To visualize the exported interaction data, we use an improved version of the *ProjectionPathExplorer* tool [23], with online embedding functionality based on tsnejs [26], umap-js [39] and Graphology ForceAtlas2 [40]. To increase the comprehensiveness of pattern recognition through both topology- and attribute-driven layout, we added a feature to show both layouts simultaneously in a multiple-coordinated view. Additionally, as outlined in Section 4.3.3, we use summary visualizations as suggested by Eckelt et al. [13], for which we implemented custom visualizations for both user studies, see Figure 5a and 10. All sessions described in this paper can be explored online[3].

6 Results

In this section, we describe patterns identified within interaction provenance data from synthetically generated sessions and discuss detailed patterns observed in two user studies with real interaction provenance data. The generated sessions illustrate the visual patterns for data types, whereas the real user sessions demonstrate the utility of *Provectories* in studying actual analysis provenance. The first user study shows the analysis of user sessions using the social-economical dataset in Gapminder. The second user study examines the analysis provenance of 12 different sessions from the study by Gadhave et al. [16], with six sessions for outlier tasks (three for outliers based on clustered data and three for outliers based on linear regression) and six sessions for clustering tasks. The supplementary material

[1] https://github.com/jku-vds-lab/sensemakingspace/
[2] https://github.com/jku-vds-lab/provectories-user-intent/
[3] Prototype: https://provectories.jku-vds-lab.at

contains Figures for all analyzed projections using different layouts, tasks, datasets, and sessions. To interactively explore the interaction provenance data, please refer to our online prototype.

6.1 Patterns for Compound Representations

The goal of the synthetically generated sessions was to study data type-specific patterns in embeddings based on compound vector representations. For the synthetic generated sessions, we use the Gapminder dataset. We started by creating sessions in which only a single data type (e.g., numerical or set) is changed in a predefined way and continued, e.g., by an alternation between a single random country selection and a random increase of a year at a sampling rate of maximum 30. Additionally, we manually created 15 sessions at a length of 30 states to verify the insights. Further information can be retrieved from the supplemental material. For the set attribute, we chose the 2-norm as a distance function, so that the average distance between two random subsets is close to 0.7, but the between typical country selections with few items is close to 0.5.

As expected, no data-type-specific patterns are visible in the topology-driven layout, while we were able to extract patterns for Boolean, categorical, numerical, and set attributes from the attribute-driven layout (see Figure 6).

Bool *Boolean* attributes are distinctly separated within the embedding space. In Figure 6a, this can be seen as the Boolean attribute *religion* and *continent* occupy their own areas in the embedding. We found that for the synthetic data, a separation of the embedding into two distinct regions almost always resulted from a Boolean attribute, if the weighting was kept equal.

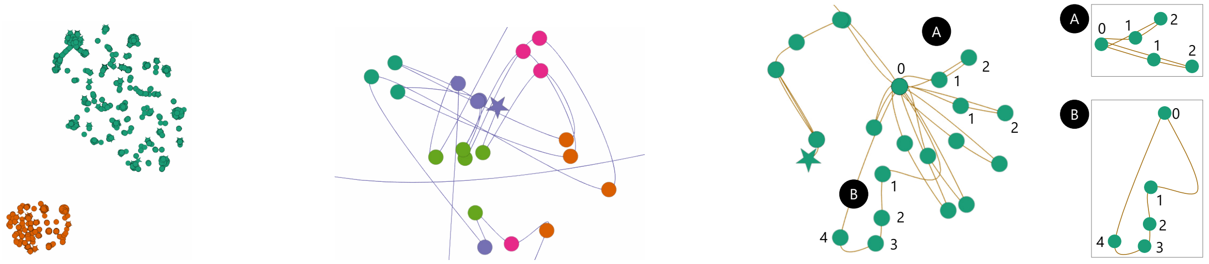
Cat Like Boolean changes, *categorical* attribute changes can cause the formation of clusters for each category in the embedding. Furthermore, certain trajectory patterns can reveal categorical changes. In Figure 6b, for instance, a cluster with the same value for the size attribute *population* is shown. Within this cluster, categorical changes in another attribute (here, x-axis) lead to substructures (*fertility rate*, *child mortality*, or *GDP*) that are connected by crossing, zigzagging lines. This phenomenon can be more pronounced by varying the weight on a respective data type, giving rise to hierarchical clustering, as explained later.

Num As shown in Figure 1❷, changes in *numerical* attributes lead to a chain of states in ascending or descending order. With regard to interaction provenance, the states need not be traversed by the user explicitly in this sequential order but the states automatically form a chain based on the definition of the numerical distance. This chain pattern is consistent for all three attribute-driven approaches (t-SNE, MDS, and UMAP).

Set If only single set items are selected in each state, all of these states have a mutual distance of $1/\sqrt{n}$, where n represents the total number of countries within the embedding space. The observation of accumulatively selecting a country can be seen in Figure 6. t-SNE, which attempts to preserve high-dimensional distances, gives circular arrangements for single selected set attributes. Thus, a combination of single and multiple selected countries leads to a ring pattern, as outlined in Figure 6c. Here, 🅐 represents (1) a single selected set attribute as inner ring, and (2) a second, added country as the outer ring, before (0) both countries are deselected again. This ring structure arises from a distance of $1/\sqrt{n}$ between states with different single-country selections and a distance of $2/\sqrt{n}$ between states with two different countries selected.

Weighting. All patterns described so far were identified for equally weighted attributes (i.e., $w_i = 1$ for all i). If the weight for an attribute type T is increased (e.g., $w_i = 10$ for some i with $k(i) = T$), the patterns related to that data type become more dominant in the attribute-driven layout. For instance, increasing the weight for a numerical attribute forces more states to be placed along a shared axis representing that numerical attribute.

Hierarchical clustering. The weighting can be adjusted to focus on a subset of data types while reducing or completely removing the effect of the other types. For the attribute-driven layout shown in Figure 7, the weight of numerical and set attributes was reduced to zero ($w_i = 0$ for all i with $k(i) = \text{num}$ or $k(i) = \text{set}$). This gives rise to a hierarchical clustering based on the remaining Boolean and categorical attribute values. Figure 7 1 shows a clear separation of the application states based on whether the color attribute represents *continent* or *religion*; 2 within the *religion* cluster, a further division is determined by the size attribute; 3 within each cluster of equal-size value, the attribute mapped to the x-axis causes a further sub-clustering. The values of these attributes are shown in the summary



(a) ***t*-SNE**—Boolean distinction over multiple sessions

(b) ***t*-SNE**—Categorical changes in multiple sessions

(c) ***t*-SNE**—Single and multiple set selections within a single session

Fig. 6: Patterns identified from synthetically generated single and multiple sessions using an attribute-driven layout. *Gapminder* data showing (a) a Boolean distinction between *religion* (●) and *continent* (●) within the embedding space; (b) categorical changes (● ● ● ● ● the colors show clusters for x-axis attributes); and (c) single and multiple set selections, where the number of selected states is indicated (0–4).

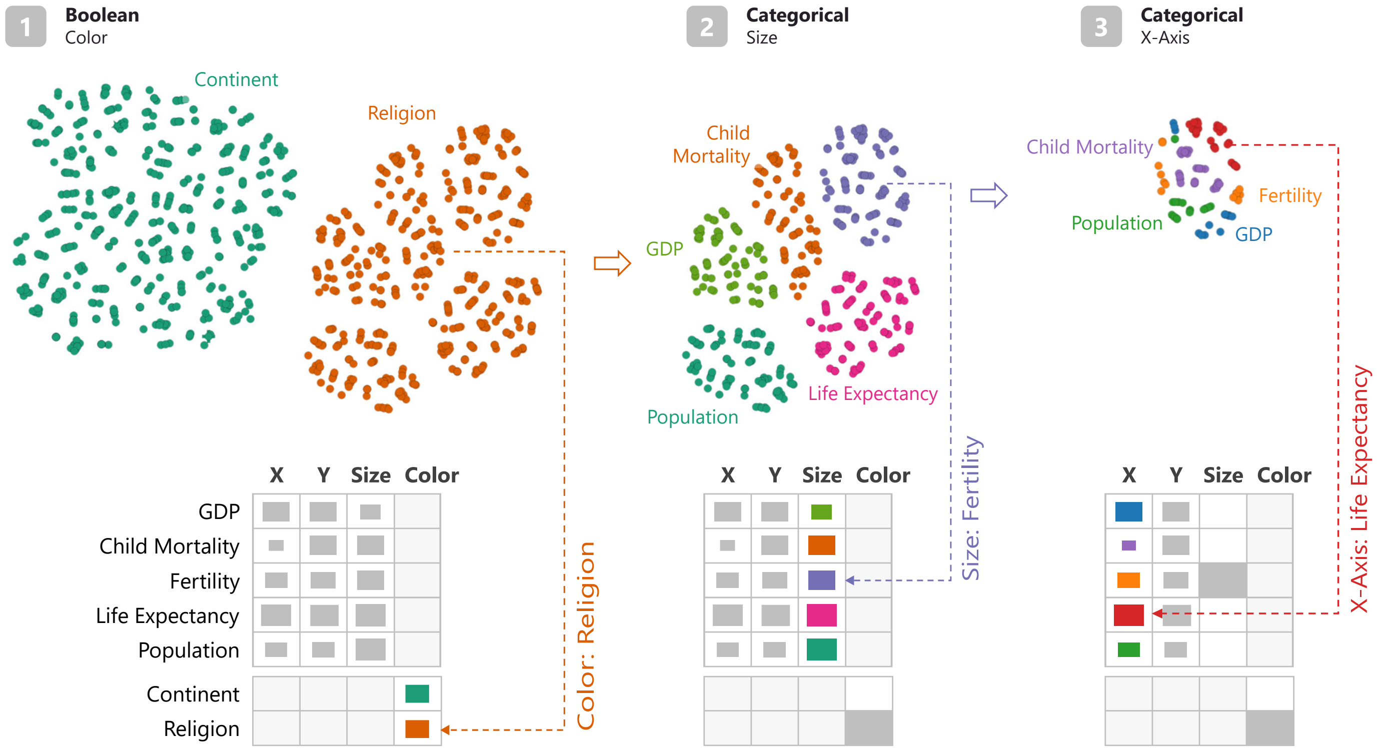


Fig. 7: Hierarchical clustering for synthetic user sessions emerges in the following order: 1 Boolean, 2 categorical, 3 categorical after removing the effect of numerical and set attributes by setting their weights to zero ($w_{\text{num,set}} = 5$). The simplified summary visualization shows the relative frequencies of the Boolean and categorical attributes in the corresponding attribute color.

visualizations in the lower part of Figure 7. Consequently, by adapting the weights, the ordering of attributes within the hierarchy can be changed.

6.2 Meta-Analysis of Gapminder User Sessions

The goal of the first user study was to confirm the patterns observed from the synthetic sessions and discover further analysis patterns from single and across multiple user sessions. We conducted a user study with 32 participants (m: 17, f: 15). The participants were students of a Data Science master program, as part of which they attended an introductory course on data visualization. We asked participants to find answers to four tasks following the Brehmer and Munzner taxonomy [4] by using the Gapminder tool.

We designed **T1** and **T2** as directed tasks (identification task), where the answers could be identified within a small number of interactions. In contrast, **T3** (comparison task) and **T4** (summarization task) are exploratory, open-ended tasks, which typically lead to longer sequences, see Table 2. As described in Section 5, we made use of the Vega-Gapminder tool that saves the interaction provenance. We asked the participants to download the interaction provenance after completing each task so the starting point for each session could be identified. We removed the sessions that contained all tasks in one file, which reduced the total number of sessions to 109.

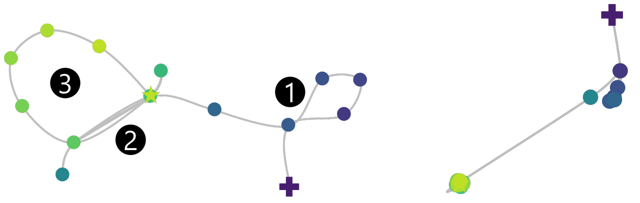
Data Types. We were able to confirm the patterns observed in the synthetic sessions for all four data types (see Section 6.1). As expected, **T1** did not reveal any data-type-specific patterns because accomplishing the task required only the year to be changed once. Since task **T2** evoked a Boolean change, sessions related to that task forms two clusters — although no longer as obvious as in the synthetic sessions. Moreover, categorical changes become apparent for both single and multiple sessions. The open-ended task **T3** requires categorical changes, numerical variations, and set alternations, where most participants set the target attribute to *population* on the x-axis or mapped it to the size. As anticipated, **T4** consisted mainly of categorical changes, where participants explored the data point distribution from

TABLE 2
Overview of the four tasks from the user experiment with average answer correctness and average number of steps taken to accomplish a task.

Task	Question	∅ Answer Correctness	∅ Number of Steps
T1	In 2015, select (a) the country with the highest GDP, and (b) the country with the largest population.	78.0 % ± 0.3 %	17.5 ± 14.0
T2	In 1843, select the Muslim country that has (a) the highest child mortality rate (b) the highest fertility rate.	87.0 % ± 0.3 %	18.7 ± 13.7
T3	Select the European country that had the largest relative drop in population between 1939 and 1945.	40.6 % ± 0.5 %	27.7 ± 23.8
T4	Select any country on the continent that has the highest correlation between any two attributes in 1945.	0.3 % ± 0.1 %	29.8 ± 34.0

the Gapminder scatterplot for almost all possible attribute combinations and selected single countries at the end of the sessions. The attribute-driven layout using t-SNE can be seen for **T4** in Figure 1, showing the interaction provenance of all users. Overall, the data type observations in all four tasks match the patterns from the synthetically generated sessions.

Analytical Strategies. **T1** was answered correctly by 78.0% of the participants using an average of 17.52 ± 14.01 steps. In both layouts, it can be seen that most participants had already found the answers to both subquestions after an average of four steps, but continued to explore the data and the tool by using the slider for the numerical attribute or the drop-down menu for categorical changes. As these additional steps are not necessary to complete the task, we call this process a *random walk*. Additionally, superimposing trajectories pointing from one cluster to another reveals that most participants chose the same analysis steps to accomplishing the task. States visualized by the attribute-driven layout distinctly show two small clusters within the embedding for both answers. In contrast, in the topology-driven layout, no unique positions for the answers can be identified. This can be attributed to the higher number of nonidentical states (e.g., attribute on x was placed on the y-axis). Thus, answers that are relatively close to the actual answer point to the outer region of the embedding if no other user selected the same application state. It is important to note that without the summary visualization, such sessions cannot be distinguished from random-walk analysis strategies. For **T2**, half of the participants started by changing the year, whereas the other half began by changing the Boolean attribute first. This can be seen by observing directed trajectories for single user sessions in the embedding with the history graph. Particularly noticeable are the variations in categorical attributes. Participants confirmed the country selection several times by changing the assignment of the target attribute to different categorical positions (e.g., x-axis, y-axis, mark size). These changes formed the zigzag pattern as shown for the synthetic sessions in Figure 6b. About one third of the participants (35.38%) completed the task by identifying both answers (country with the highest child mortality and fertility rate) in the same application state with both (T2a) *child mortality* and (T2b) *fertility* as categorical axis options.



(a) **Force-Directed** layout (b) t-**SNE** layout

Fig. 8: Gapminder user study: single user session with 23 steps from **T3**, where ❶ shows alternation between y-axis items, ❷ toggling between two states and ❸ a verification loop by screening for incorrect states. The user session starts at ✚ and ends at ★.

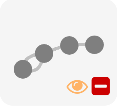
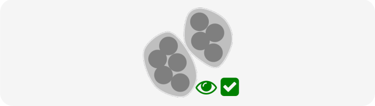
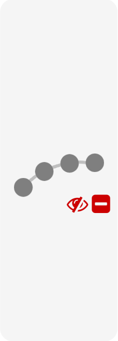
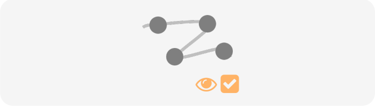
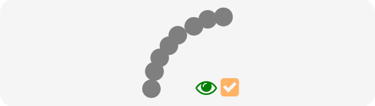
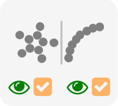
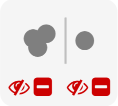
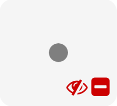
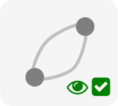
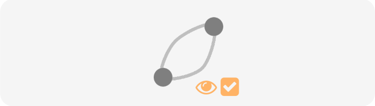
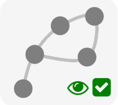
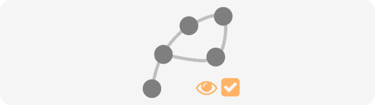
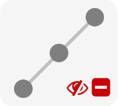
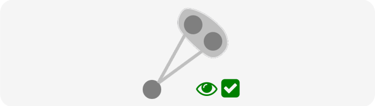
We observed that multiple exploration paths lead to the correct answer. For **T3**, 18 out of 22 sessions first converge on one unique application state before continuing the analysis in various ways. Participants started by changing the year to 1940 before selecting different countries and varying categorical characteristics to determine the largest relative drop in the population between 1939 and 1945. As shown in Figure 8, one user, for instance, started (✚) by assessing the correlation between two attributes by ❶ selecting all y-axis attributes (except for child mortality), and then revisited the initial attribute. This sequence leads to a visual loop in topology-driven layout. The user continued by changing the year and ❷ alternating between two set attributes. Toggling compares the two alternatives and verifies the final selection. Before terminating the analysis, the participant ❸ looked at four other countries and confirmed the initial selection (★). **T3** also benefits from visualizing the interaction provenance with the attribute-driven layout as participants selected different countries after changing the year to 1940, and the selected states are positioned close to each other. Hence, the topology-driven layout treats these states as independent and unique (see Figure 8a), and the attribute-driven approach emphasizes the similarity of the application states for different analysis processes (see Figure 8b). **T3** has an average answer correctness of 40.6% and an average number of step taken to accomplish the task of 27.27 ± 23.8.

The last exploration task **T4** shows identical states and overlapping trajectories within the embedding with **T3** because the same year—1945—needed to be selected to accomplish the task. In general, both open-ended tasks cover a large area within the embedding space. Furthermore, **T4** shows an average response accuracy of 0%, while the number of steps to accomplish the task is higher (29.8 ± 34.0) than for the directed tasks (**T1** and **T2**). Overall, participants tried to find the highest correlation between any two attributes by varying all attribute combinations for any categorical combinations for **T4**. Furthermore, the attribute-driven layout for **T4** shows a zigzag pattern, which means that x- and size attributes are contained within clusters of y-attributes, which confirms the hierarchical dependency based on categorical attributes (see Section 6.2).

Layout Applicability. Based on the insights gained from synthetically generated sessions and real user interaction provenance, we summarize the identifiable patterns in Table 3. Dependent on the layout and the visual pattern, we introduce an indicator that describes the readability and validity of each pattern. The former indicates whether it is possible to identify this pattern within the embedding space, and the latter signifies the reliability of the pattern.

Data-type-specific patterns emerge only in an attribute-driven layout. Each Boolean item occupies its own area within the embedding space, which leads to two distinct areas for all three techniques (t-SNE, MDS, and UMAP); this can be accentuated by putting a higher weight on the data type. When only a single session is embedded by itself, categorical changes are difficult to extract due to the low number of states within the embedding space. In contrast, when multiple sessions are embedded at the same time, additional states provide enough context that clusters for each attribute can emerge. However, based on the high number of trajectories, visual clutter can result. In these cases, making use of adaptations in the visual encodings with the Projection Space Explorer supports the identification of cate-

TABLE 3
Visual and data patterns extracted from various techniques, with indicators for the levels of readability (👁̸ low, 👁 medium, 👁 high) and validity (⊟ low, ☑ medium, ☑ high).

Patterns	Topology-driven	Attribute-driven		
		t-SNE	MDS	UMAP
Bool				
Cat				
Num				
Set				
Comparison $s_i \leftrightharpoons s_{i+1}$				
Looping $s_i = s_{i+k}$				
Similar selection $s_i \approx s_j$				

gorical changes by filtering and therefore visually removing non-relevant user sessions or applying a color encoding based on a specific attribute. Further, tracing single sessions using the history feature highlights the session of interest by outlining differences between two consecutive states. In contrast, a chain of states emerging from numerical value changes has a high validity within all three algorithms; also cumulatively selected set items resemble this behavior. In addition, single selected set attributes form a circular state pattern for *t*-SNE. MDS and UMAP do not yield a clear pattern, since many data points converge to almost a single position in the embedding space. Although *t*-SNE is known for preserving local structures better, while UMAP is said to preserve global structures better, the attribution of set patterns was the only clear difference we could identify between both approaches. We also observed—as expected—that with increasing perplexity values the *t*-SNE scatterplots tended to resemble those constructed with UMAP.

To trace users' analysis steps, both topology-driven and attribute-driven layouts can be applied to identify steps revisited in single and multiple sessions based on the removal of duplicates (and, similarly, for loops containing intermediate states). However, the readability and validity is higher in topology-driven layouts. Thus, confirmation or verification tasks can be observed for single user sessions. Due to overlapping and intersecting trajectories, the identification of an analytical reasoning process for a single session becomes more difficult with an increasing number of sessions. We address this shortcoming with the history graph (see Figure 5b) that allows meta-analysts to detect and understand patterns of single user sessions in multiple simultaneously displayed sessions by highlighting the session of interest. Near identical data points can only be identified in an attribute-driven layout, where they are positioned closer to each other. Consequently, overlapping trajectories signify application states that were also visited by other users in the same analytical sequence. In MDS and UMAP, however, data points of set attributes almost overlap in the embedding space, whereas the chain pattern of numerical values results in a small distance between similar data points. For UMAP, this may be improved by choosing a different setting of the "mindist" parameter.

Further, individual analysis steps or steps of a random walk represent unique data points. Particularly for MDS and UMAP, Boolean, and categorical changes evoke visually distant data points. Based on the entropy of the embedding space, individual data points or even sessions become distinctive. Moreover, sessions that end at a unique state composition are positioned at the outer part of the embedding in the topology-driven layout.

6.3 Meta-Analysis of User Intent Sessions

We demonstrate the general applicability of *Provectories* using interaction provenance data from the users' intent study by Gadhave et al. [16]. They conducted a crowdsourcing user study with 130 participants, where each participant conducted five different tasks. Among these tasks, participants were asked to select outliers or data points that belong to a cluster in a scatterplot. They analyzed two conditions. In the first condition, users were computationally supported by an auto-complete feature to select, e.g., the desired outliers. This feature became apparent after selecting the first data point. For the second condition, users had to accomplish the tasks manually without any computational assistance. In total, 12 different datasets were used for outlier tasks (cluster and linear) and six for cluster tasks, each with three difficulty levels (easy, medium, hard).

To analyze the user behavior, we extracted the set of selected data points after each interaction. Unlike in the Gapminder example, we chose not to use this information as a simple set attribute, but instead calculate a more meaningful feature vector that concisely describes both the number and the position of all selected points. We first normalize the coordinates of all data points from the different datasets that users interacted with. We then construct a 10-by-10 grid and count the number of selected points within each grid cell. The resulting 2D histogram is flattened into a vector and the vectors are compared using the cosine similarity. This encoding ensures that point selections in similar regions of the scatterplots are close together, even if the sets of selected points do not match exactly. It also enables meaningful comparison of user selections across different datasets. For the summary visualizations, we simply show scatterplots of the selected points, with opacity encoding in how many of the analyzed states a given point is part of the selection.

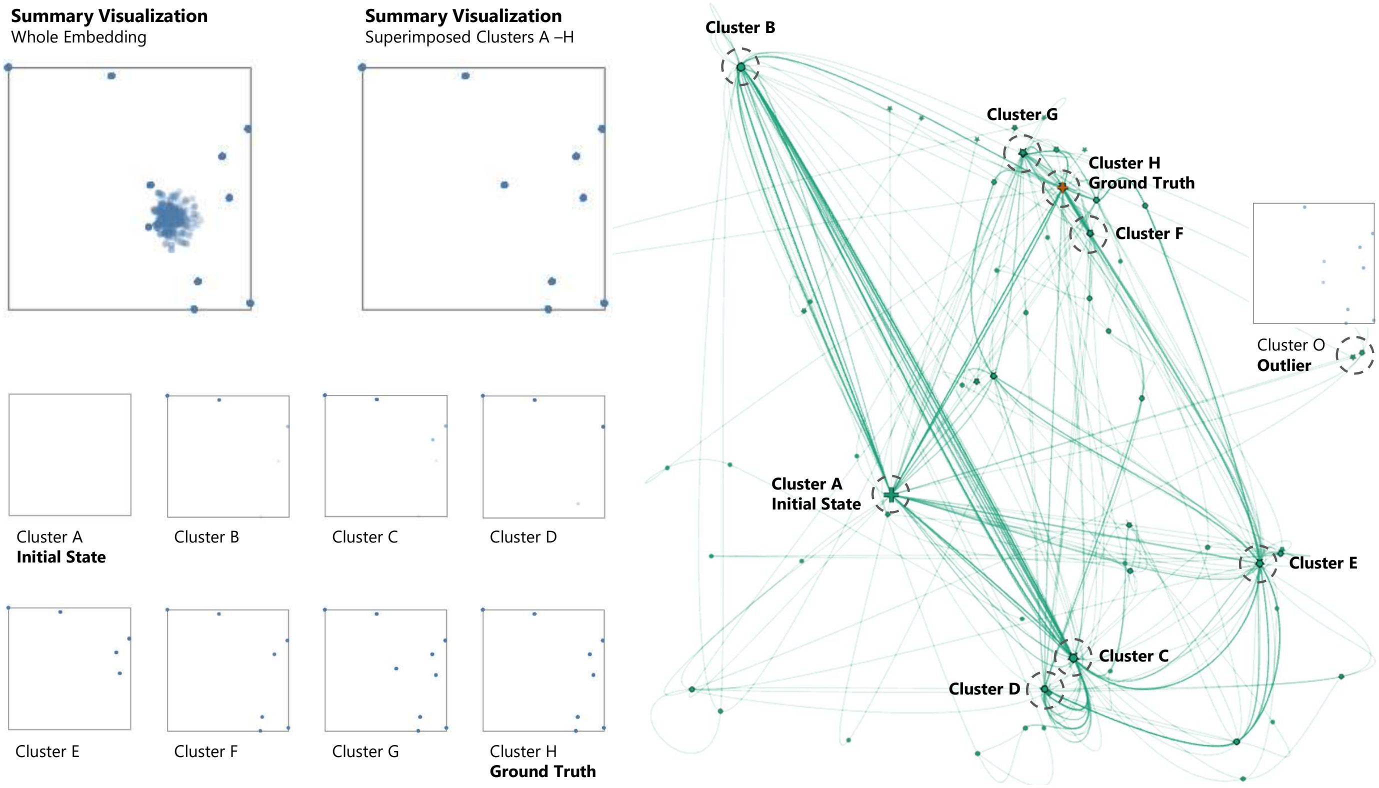


Fig. 9: Cluster-based analysis using t-SNE on the example of the *outlier (cluster) medium 2* dataset performing a multiple-user investigation. Distinct clusters (Cluster A–H) can be observed for outlier selections and superimposed trajectories indicating that the selection of the data points were performed in the same/similar sequence by multiple users.

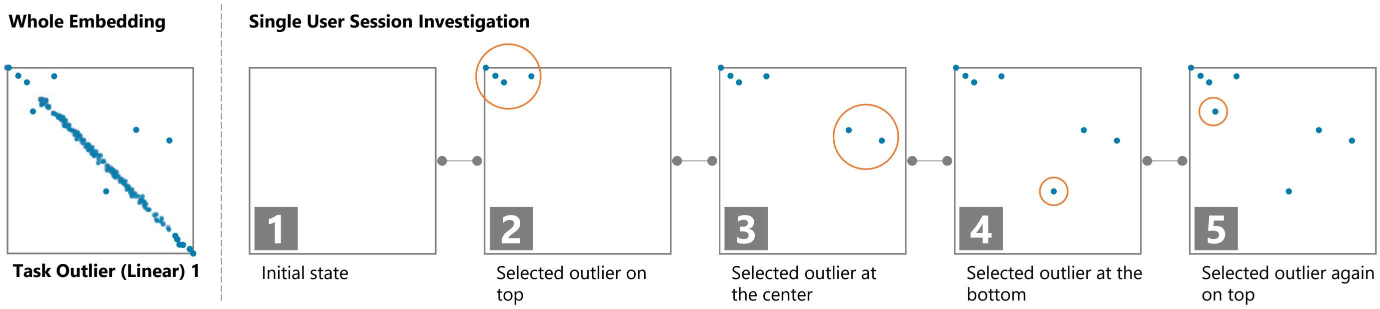
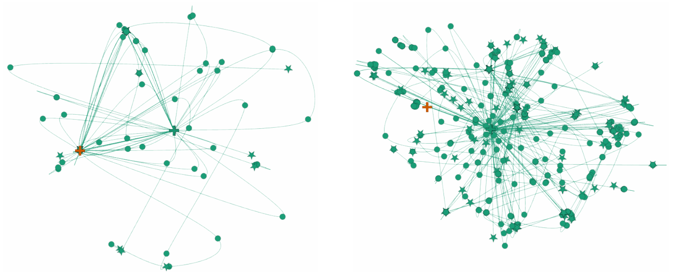


Fig. 10: On the left, summary visualization for all interaction provenance states within the embedding for the dataset *outlier (linear) easy 1*. 1—5 shows the summary visualizations using the storytelling feature (playback features as within Nobre et al. [35]) for analyzing a single user session. Outlier were first selected on top before selecting data points at the bottom of the scatterplot and going up again.

We additionally enrich the *Provectories* visualization with meta-attributes to understand the embeddings in more detail, following the high-dimensional data summary visualization from Eckelt et al. [13]. Meta-attributes for this user study are the user ID, the task ID, the accuracy per task on a user level, the task difficulty level, the Boolean attribute of auto-complete used, and the selected rank of the prediction that was used by a user.

Analytical Strategies. In line with the observed *select and refine* analysis strategy identified by Nobre et al. [35], we can confirm this strategy for single user sessions by means of the playback function in the history graph. Moreover, when performing single-session investigations for the outlier tasks, a top-to-bottom approach becomes visible (**S3** and **S4**). Thus, participants primarily started to select the outliers at the top of the scatterplot before selecting outliers towards the center or bottom, see summary visualizations in Figure 10. This analysis strategy can also be observed for multiple users, where outlier selections form distinct clusters within the embedding (**M3**), see cluster A–H in Figure 9. In contrast, Cluster O is located away from these clusters. Here, the summary visualization reveals that the most prominent of all outliers, the point in the upper left corner of the scatterplot, was not selected. The ground truth is colored in orange (✚). Hence, the summary visualization for all states in the embedding shows almost the original scatterplot from the dataset, whereas the summary visualization only for the visually identifiable clusters in the embedding provides information about the outlier coordinate positions. In addition, through both the transparency of trajectories but also the direction in which the trajectories are pointing to, it can be seen that the selection of outliers was performed by multiple users in the same/similar sequence.



(a) Cluster Easy 1

(b) Cluster Hard 1

Fig. 11: Projections for an easy and a hard cluster task. Easy tasks show superimposed trajectories and few states. The hard task shows numerous unique states and trajectories pointing in all directions. The ground truth (✚) is colored in orange.

Dataset Difficulty. Further, in line with the performance measures from Gadhave et al. [16], *Provectories* shows different patterns in the embedding for the level of difficulty per dataset. Easy tasks show numerous superimposed trajectories pointing from one state to another, indicating states with the same data point selections (see dataset *cluster easy 1* in Figure 11a). In contrast, hard tasks show a higher number of unique states in the embedding and hardly any overlapping trajectories (see dataset *cluster hard 1* in Figure 11b) (**M1**). Also as outlined in Figure 11, the ground truth state colored in orange was found exactly for the easy task, whereas a 100% answer correctness was not reached by any user for the *cluster hard 1* dataset. Gadhave et al. also assumed a confidence interval of 95% to analyze the answer correctness. When looking at clusters close to the ground truth state, we can visually determine distinct cluster positions in the embedding for the easy tasks, whereas the auto-complete prediction from the user intent study for hard tasks does not show any clear clusters, see supplementary material. We see this observation as a potential approach for usability testing of a system by assessing not only the number of click events a user needs to accomplish a task, but rather to see if multiple users followed the same path to navigate through a tool. This could help meta-analysts to identify aspects of the tool that might distract the user.

Prediction. After consultations with the authors, it remained unclear if the computational auto-complete suggestion directly led the users to the final answer or if they continued their analysis process. Thus, we investigated the "supported" condition in more detail. By using color and shape encodings in addition to the summary visualizations, we were able to observe that 38.7% of the participants who used the prediction feature for outlier tasks, finalized their session on the ground truth state—52.5% for the outlier (cluster) and 25.5% of the outlier (linear) task. The descriptive values could be extracted from the summary visualization. For the cluster tasks, only 21.8% of the users selected the correct suggested prediction (three prediction options were given to the user) and therefore ending their session on the ground truth state. Hardly any participant performing a task on any hard dataset (outlier and cluster) reached the ground truth state–except for the *outlier (cluster) hard 2* dataset. Moreover, *Provectories* enabled the identification of sessions in which users selected incorrect predictions and consequently refined the data point selection for the scatterplot (**M2**).

By analyzing the interaction provenance of the second user study, analysis strategies such as top-to-bottom, select and refine, but also the identification of "dead ends", which are resolved by the auto-complete feature, could be observed. Again, "dead ends" or looping behaviors could be best observed with the topology-driven layout. By comparing multiple users performing the same task on a dataset, it was possible to identify clusters in the attribute-driven layouts, which indicate the same or similar state constitution. By taking superimposed trajectories and the direction of the trajectories into account, solving strategies can be determined.

7 Discussion

Vector Representations. In a few visual analytics applications, users manipulate an object for which a vector representation is readily available (e.g., an underlying machine learning model). We thus see the compound representation explained in Section 4.2—and showcased with the Gapminder user sessions—as a potential starting point for provenance meta-analysis. For many applications it may be possible to encode each interactive visual component based on the data type of its underlying variable. We showed that the resulting compound representations lead to certain type-specific patterns in the *Provectories* visualization, which may be accentuated through weighting. However, care must be taken to correctly interpret patterns under the effect of hierarchical clustering (see Section 6.2). In our meta-analysis of the user intent study data [16] presented in Section 6.3, we show the potential of a semantic state representation not directly based on low-level variables. Unfortunately, it is difficult to make general statements about such representations, as they need to be constructed on a case-by-case basis, with close consideration of the artifacts manipulated by the users and the tasks performed.

One aspect that is not considered in the layout calculation—apart from the sequential order for drawing the lines—is the time users took between states. We refrained from adding the timestamp or time differences to the state representations as this would introduce differences between otherwise equivalent states, which could in turn obfuscate the patterns that we identified. However, it would still be interesting to explore the information contained in the time data. We see two potential ways going forward: (1) adding additional encoding options for line segments, which could be used to identify slow and fast stages of the user sessions; and (2) incorporating the dimension of time separately, in way similar to how we process connectivity information in our hybrid layout approach proposed below.

We found it especially challenging to find suitable state representations in cases where users can create unlimited visual components themselves (e.g., new views in dashboards). We hypothesize that for such tools a representation similar to Fock states in quantum mechanics [15] could be used to describe the components in the infinitely large configuration space. In such a representation, instead of listing all views with their attributes, the possible attribute combinations are listed along with associated "counts" of views that share those attributes.

Motifs. With our novel visual analysis approach, we extracted patterns based on the connectivity and similarity of application states. To increase the knowledge about analysis sequences and to reduce visual complexity, we suggest using a motif-based aggregation for both layout approaches [48]. Detection of motifs allows us to aggregate the provenance graph or parts thereof while preserving the high-level structure. This adds the potential of chunking interaction sequences for a more compact display. Furthermore, identified patterns could be rendered as a sequence of actions and compared across multiple sessions. In line with the idea of using a motif-based approach, edge bundling could be applied to reduce the visual clutter. This could be particularly of benefit when visualizing multiple user sessions in the same embedding space due to clutter emerged through overlapping trajectories.

Hybrid Layout Approach. To combine the advantages of both the topology *and* the attribute-driven layout, we have started to develop a hybrid layout approach. In the purely attribute-driven layout used in our work, the distance matrix for t-SNE or UMAP is calculated directly from the attribute values (see Section 4.2), while the topology-driven layout is based on the connectivity of states. Our hybrid approach builds on *tsNET* [28], which creates a topology-driven layout by transforming the adjacency matrix of a graph to a distance matrix which is then used for t-SNE. We combine this topology-driven distance matrix with the attribute-driven one and use their weighted sum for a hybrid embedding. For the sessions from the user intent data, we found that a hybrid embedding with low weight on the attribute-based distances reveals similar patterns as the purely attribute-driven one, while circumventing the shortcoming of the degenerate distances for empty selections. We believe that the applicability of such hybrid layouts exceeds the scope of *Provectories*, and we want to further refine and study this technique in future work.

8 Conclusion

In this paper, we have presented a novel visual analysis approach to extract patterns from interaction provenance data. Our *Provectories* approach consists of three steps: (1) the acquisition of interaction provenance data in the form of logged application states, (2) the construction of feature vectors representing these states, and (3) the visualization of provenance using topology- and attribute-driven layouts. By interactively exploring such visualizations for compound representations and real user sessions, patterns based on data types and analytical reasoning processes can be revealed. We demonstrate our approach by means of two user studies and were able to increase the comprehension of interaction logs using *Provectories*. However, interaction provenance from other applications, in particular, feature-rich tools such as Tableau and Power BI remain to be explored. We strongly believe that *Provectories* can fill a gap in the field of provenance and sense-making to improve understanding of similarities between analysis processes and user-specific behaviors.

Acknowledgments

We would like to thank Kiran Gadhave and Alexander Lex for providing us with interaction provenance data for the second use case and for being helpful with questions. This work has been supported by the FFG, Contract No. 881844: "Pro^{2}Future is funded within the Austrian COMET Program Competence Centers for Excellent Technologies under the auspices of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology, the Austrian Federal Ministry for Digital and Economic Affairs and of the Provinces of Upper Austria and Styria. COMET is managed by the Austrian Research Promotion Agency FFG." Additional support was granted by the State of Upper Austria and the Austrian Federal Ministry of Education, Science and Research via the LIT – Linz Institute of Technology (LIT-2019-7-SEE-117), and by the State of Upper Austria (Human-Interpretable Machine Learning).

References

[1] Anaconda Inc. Datashader. Accurately render even the largest data, 2016. https://datashader.org/.

[2] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):559–568, 2016.

[3] T. Blascheck, F. Beck, S. Baltes, T. Ertl, and D. Weiskopf. Visual analysis and coding of data-rich user behavior. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 141–150. IEEE, 2016. ZSCC: 0000016. doi: 10.1109/VAST.2016.7883520

[4] M. Brehmer and T. Munzner. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2376–2385, 2013. doi: 10.1109/TVCG.2013.124

[5] E. T. Brown, A. Ottley, H. Zhao, Q. Lin, R. Souvenir, A. Endert, and R. Chang. Finding Waldo: Learning about Users from their Interactions. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1663–1672, 2014. doi: 10.1109/TVCG.2014.2346575

[6] E. T. Brown, S. Yarlagadda, K. A. Cook, A. Endert, and R. Chang. ModelSpace: Visualizing the Trails of Data Models in Visual Analytics Systems. In *MLUI 2018: Machine Learning from User Interactions for Visualization and Analytics*, p. 11. IEEE, 2018.

[7] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo. VisTrails: visualization meets data management. In *Proceedings of the international conference on Management of data (SIGMOD '06)*, p. 745. ACM Press, 2006. doi: 10.1145/1142473.1142574

[8] H. Chung, S. Yang, N. Massjouni, C. Andrews, R. Kanna, and C. North. VizCept: Supporting synchronous collaboration for constructing visualizations in intelligence analysis. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pp. 107–114. IEEE, 2010. doi: 10.1109/VAST.2010.5652932

[9] P. Cowley, L. Nowell, and J. Scholtz. Glass Box: An Instrumented Infrastructure for Supporting Human Interaction with Information. In *Proceedings of the Conference on System Sciences (HICSS '05)*, p. 296c, 2005. doi: 10.1109/HICSS.2005.286

[10] Z. T. Cutler, K. Gadhave, and A. Lex. Trrack: A Library for Provenance Tracking in Web-Based Visualizations. In *IEEE Visualization Conference (VIS)*, pp. 116–120. IEEE, Salt Lake City, UT, USA, 2020. ZSCC: 0000003. doi: 10.1109/VIS47514.2020.00030

[11] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Recovering reasoning process from user interactions. *IEEE Computer Graphics & Applications*, 29(3):52–61, 2009.

[12] C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, and G. Robertson. GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks While Supporting Exploration History. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, pp. 1663–1672. ACM, 2012. doi: 10.1145/2207676.2208293

[13] K. Eckelt, A. Hinterreiter, P. Adelberger, C. Walchshofer, V. Dhanoa, C. Humer, M. Schöfl, C. Steinparz, and M. Streit. Visual Exploration of Relationships and Structure in Low-Dimensional Embeddings. preprint, Open Science Framework, Apr. 2021. doi: 10.31219/osf.io/ujbrs

[14] M. Feng, E. Peck, and L. Harrison. Patterns and Pace: Quantifying Diverse Exploration Behavior with Visualizations on the Web. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):501–511, 2019. doi: 10.1109/TVCG.2018.2865117

[15] V. A. Fock. Konfigurationsraum und zweite Quantelung. *Zeitschrift für Physik*, 75(9–10):622–647, 1932. doi: 10.1007/BF01344458

[16] K. Gadhave, J. Görtler, Z. Cutler, C. Nobre, O. Deussen, M. Meyer, J. M. Phillips, and A. Lex. Predicting intent behind selections in scatterplot visualizations. *Information Visualization*, 20(4):207–228, Oct. 2021. Publisher: SAGE Publications. doi: 10.1177/14738716211038604

[17] M. Gleicher. Considerations for Visualizing Comparison. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):413–423, 2018. doi: 10.1109/TVCG.2017.2744199

[18] S. Gomez and D. Laidlaw. Modeling task performance for a crowd of users from interaction histories. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '12)*, p. 2465. ACM Press, 2012. doi: 10.1145/2207676.2208412

[19] D. Gotz and M. X. Zhou. Characterizing Users' Visual Analytic Activity for Insight Provenance. *Information Visualization*, 8(1):42–55, 2009. tex.ids: gotz_characterizing_2009-1. doi: 10.1057/ivs.2008.31

[20] J. C. Gower. A General Coefficient of Similarity and Some of Its Properties. *Biometrics*, 27(4):857, Dec. 1971. doi: 10.2307/2528823

[21] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit. From Visual Exploration to Storytelling and Back Again. *Computer Graphics Forum*, 35(3):491–500, 2016. doi: 10.1111/cgf.12925

[22] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '08)*, 14(6):1189–1196, 2008. doi: 10.1109/TVCG.2008.137

[23] A. Hinterreiter, C. Steinparz, M. Schöfl, H. Stitz, and M. Streit. Projection Path Explorer: Exploring Visual Patterns in Projected Decision-Making Paths. *ACM Transactions on Interactive Intelligent Systems*, 11(3–4):Article 22, 2021. doi: 10.1145/3387165

[24] J. Hollan, E. Hutchins, and D. Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction*, 7(2):174–196, 2000. doi: 10.1145/353485.353487

[25] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLoS ONE*, 9(6):e98679, 2014. doi: 10.1371/journal.pone.0098679

[26] A. Karpathy. tSNEJS, 2014. https://github.com/karpathy/tsnejs/.

[27] N. Kerracher, J. Kennedy, and K. Chalmers. A Task Taxonomy for Temporal Graph Visualisation. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2015. doi: 10.1109/TVCG.2015.2424889

[28] J. F. Kruiger, P. E. Rauber, R. M. Martins, A. Kerren, S. Kobourov, and A. C. Telea. Graph Layouts by t-SNE. *Computer Graphics Forum*, 36(3):283–294, 2017. doi: 10.1111/cgf.13187

[29] Z. Liu and J. Stasko. Mental Models, Visual Reasoning and Interaction in Information Visualization: A Top-down Perspective. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):999–1008, 2010. doi: 10.1109/TVCG.2010.177

[30] K. Madanagopal, E. D. Ragan, and P. Benjamin. Analytic Provenance in Practice: The Role of Provenance in Real-World Visualization and Data Analysis Environments. *IEEE Computer Graphics and Applications*, 39(6):30–45, Nov. 2019. doi: 10.1109/MCG.2019.2933419

[31] W. C. Mankowski, P. Bogunovich, A. Shokoufandeh, and D. D. Salvucci. Finding canonical behaviors in user protocols. In *Proceedings of the conference on Human factors in computing systems (CHI'09)*, p. 1323. ACM Press, 2009. doi: 10.1145/1518701.1518900

[32] Margit Pohl and Johanna Doppler Haider. Sense-making Strategies for the Interpretation of Visualizations—Bridging the Gap between Theory and Empirical Research. *Multimodal Technologies and Interaction*, 1(3):16, 2017. doi: 10.3390/mti1030016

[33] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426 [cs, stat]*, 2018.

[34] P. H. Nguyen, K. Xu, A. Wheat, B. L. W. Wong, S. Attfield, and B. Fields. SensePath: Understanding the Sensemaking Process Through Analytic Provenance. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):41–50, 2016. doi: 10.1109/TVCG.2015.2467611

[35] C. Nobre, D. Wootton, Z. Cutler, L. Harrison, H. Pfister, and A. Lex. reVISit: Looking Under the Hood of Interactive Visualization Studies. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–13. ACM, Yokohama Japan, May 2021. doi: 10.1145/3411764.3445382

[36] L. G. Nonato and M. Aupetit. Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2650–2673, 2019. doi: 10.1109/TVCG.2018.2846735

[37] A. Pandey, U. Syeda, and M. Borkin. Towards Identification and Mitigation of Task-Based Challenges in Comparative Visualization Studies. *OSF Preprints*, 2020. doi: 10.31219/osf.io/5p73v

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[39] People+AI Research (PAIR) Initiative. UMAP-JS, 2019. https://github.com/PAIR-code/umap-js/.

[40] G. Plique. Graphology ForceAtlas2, 2016. https://github.com/graphology/graphology-layout-forceatlas2.

[41] M. Pohl, M. Smuc, and E. Mayr. The User Puzzle - Explaining the Interaction with Visual Analytics Systems. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2908–2916, 2012. doi: 10.1109/TVCG.2012.273

[42] P. G. Poličar, M. Stražar, and B. Zupan. openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding. *bioRxiv*, 2019. doi: 10.1101/731877

[43] E. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE Transactions on Visualization and Computer Graphics (VAST '15)*, 22(1):31–40, 2016. doi: 10.1109/TVCG.2015.2467551

[44] P. E. Rauber, S. G. Fadel, A. X. Falcão, and A. C. Telea. Visualizing the Hidden Activity of Artificial Neural Networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):101–110, 2017. doi: 10.1109/TVCG.2016.2598838

[45] H. Rosling and Z. Zhang. Health advocacy with Gapminder animated statistics. *Journal of Epidemiology and Global Health*, 1(1):11, 2011. doi: 10.1016/j.jegh.2011.07.001

[46] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A Natural Language Interface for Visual Analysis. In *Proceedings of the Symposium on User Interface Software and Technology*, pp. 365–377. ACM, 2016. doi: 10.1145/2984511.2984588

[47] H. Stitz, S. Gratzl, H. Piringer, T. Zichner, and M. Streit. KnowledgePearls: Provenance-Based Visualization Retrieval. *IEEE Transactions on Visualization and Computer Graphics (VAST '18)*, 25(1):120–130, 2019. doi: 10.1109/TVCG.2018.2865024

[48] H. Stitz, S. Luger, M. Streit, and N. Gehlenborg. AVOCADO: Visualization of Workflow–Derived Data Provenance for Reproducible Biomedical Research. *Computer Graphics Forum*, 35(3):481–490, 2016. doi: 10.1111/cgf.12924

[49] S. van den Elzen, D. Holten, J. Blaas, and J. van Wijk. Reducing Snapshots to Points: A Visual Analytics Approach to Dynamic Network Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):1–10, 2016. doi: 10.1109/TVCG.2015.2468078

[50] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[51] J. Wei, Z. Shen, N. Sundaresan, and K.-L. Ma. Visual cluster exploration of web clickstream data. In *IEEE conference on Visual Analytics Science and Technology (VAST'12)*, pp. 3–12. IEEE, 2012. doi: 10.1109/VAST.2012.6400494

[52] J. Wood, A. Kachkaev, and J. Dykes. Design Exposition with Literate Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):759–768, 2019. ZSCC: NoCitationData[s0]. doi: 10.1109/TVCG.2018.2864836

[53] K. Xu, A. Ottley, C. Walchshofer, M. Streit, R. Chang, and J. Wenskovitch. Survey on the Analysis of User Interactions and Visualization Provenance. *Computer Graphics Forum*, 39(3):757–783, 2020. doi: 10.1111/cgf.14035

[54] Yang Chen, Jing Yang, and W. Ribarsky. Toward effective insight management in visual analytics systems. In *2009 IEEE Pacific Visualization Symposium*, pp. 49–56. IEEE, Beijing, China, 2009. doi: 10.1109/PACIFICVIS.2009.4906837

[55] J. Zhao, M. Glueck, P. Isenberg, F. Chevalier, and A. Khan. Supporting Handoff in Asynchronous Collaborative Sensemaking Using Knowledge-Transfer Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):340–350, 2018. ZSCC: NoCitationData[s0]. doi: 10.1109/TVCG.2017.2745279

Conny Walchshofer is a PhD student at the Institute of Computer Graphics at Johannes Kepler University Linz, Austria. In her prior research, she focused on the perception and handling of mutlidimensional visualizations. She applies an interdisciplinary approach to judge cognitive load during the interpretation of visual representations by using physiological measurement methods (e.g., eye-tracking, heart rate variability).

Andreas Hinterreiter is a PhD student at the Institute of Computer Graphics, Johannes Kepler University (JKU) Linz. His research interests include dimensionality reduction and explainable AI. He recently spent a year at the Biomedical Image Analysis Group at Imperial College London. He received his Diplomingenieur (MSc) in Technical Physics from JKU.

Kai Xu is an Associate Professor in Data Analytics at the Middlesex University in London, UK. He has over 15 year experience in data visualization and analytics research in both the academic and industry context. Recently he has been working with the UK government departments and leading defence companies on using Visual Analytics to address the sensemaking challenges they face in (big)data analysis. His research interests include data visualization, provenance, sensemaking, and machine learning, with a focus on integrating human and machine intelligence. His work has won a few international data visualization awards.

Holger Stitz is a senior researcher at datavisyn. He finished his PhD at the Johannes Kepler University Linz in 2019. His main research interest include biomedical data visualization, provenance, and sensemaking. For more information see http://holgerstitz.de.

Marc Streit is a Full Professor of Visual Data Science at the Johannes Kepler University Linz, Austria. He finished his PhD at the Graz University of Technology in 2011. His scientific areas of interest include visualization, visual analytics, and biological data visualization. He won multiple best paper and runner-up awards at InfoVis, EuroVis, BioVis, and CHI. Marc is also cofounder and CEO of datavisyn, a spin-off company that develops data visualization solutions for applications in pharmaceutical and biomedical R&D. For more information see http://marc-streit.com.