A Bidirectional Pipeline for Semantic Interaction

Michelle Dowling*, John Wenskovitch†, Peter Hauck‡, Adam Binford§, Nicholas Polys¶, Chris North‖
Virginia Tech

ABSTRACT

Semantic interaction techniques in visual analytics tools allow analysts to indirectly adjust model parameters by directly manipulating the visual output of the models. Many existing tools that support semantic interaction do so with a number of similar features, including using a set of mathematical models that are composed within a pipeline, having a semantic interaction be interpreted by an inverse computation of one or more mathematical models, and using an underlying bidirectional structure within the pipeline. We propose a new visual analytics pipeline that captures these necessary features of semantic interactions. To demonstrate how this pipeline can be used, we represent existing visual analytics tools and their semantic interactions within this pipeline. We also explore a series of new visual analytics tools with semantic interaction to highlight how the new pipeline can represent new research as well.

Index Terms: Human-centered computing—Visualization—Visualization systems and tools; Human-centered computing—Interaction design—Interaction design process and methods

1 INTRODUCTION

Semantic interaction is a powerful interaction methodology, allowing analysts to explore and discover relationships in data [10]. Semantic interaction exploits intuitive interactions to manipulate underlying model-level parameters. Through semantic interactions such as the direct manipulation of visualizations, visual analytics tools are able to learn about the analyst's reasoning process. This learning is expressed as alterations to the parameters of underlying mathematical models, ultimately resulting in an updated visualization to reflect this newly-learned information [13]. This coupling of machine learning algorithms with the analyst's knowledge and interactions allows for the creation of robust analytics tools that collaboratively exploit the skills of both human and computer.

For example, a number of semantic interaction tools and techniques have been developed that make use of a visual "proximity $\approx$ similarity" metaphor to map observations[1] into a visualization [4, 12–14, 19, 25]. As analysts manipulate the observations in these visualizations, they communicate a desired similarity between a subset of observations, which in turn updates the underlying model parameters that define the visualization. These interactions allow an analyst to continue exploring and understanding relationships in the data without pausing to manipulate model parameters manually. This frees the analyst's cognition to focus on high-level analysis concepts rather than low-level parameter details [11]. As the analyst continues to perform such semantic interactions, the tool learns more about the analyst's reasoning, and the visualization incrementally adjusts to reflect the current data exploration [12].

Though a number of tools that use semantic interactions have been developed, each is described in a distinct manner to highlight the purpose for which the tool was built. Although there are more generalized pipelines to describe the concepts behind such visual analytics tools, such as the those proposed by Card et al. for information visualization [6] and Keim et al. [16] for visual analytics tasks, they do not incorporate sufficient focus on the interactions to fully capture the power and complexity of semantic interaction. Thus, it can be difficult to understand how semantic interactions affect the underlying mathematical models and how these mathematical models work together in a single tool.

To address this need for capturing the complexity involved in semantic interactions for visual analytics tools, we begin by exploring the characteristics of semantic interactions in such tools. With these characteristics to guide us, we define a new pipeline that can properly communicate how the visualization is created and how semantic interactions are interpreted. We then demonstrate this new pipeline's capabilities by discussing the pipeline representations for a set of existing tools as well as a selection of new visual analytics tools. Finally, we discuss other implications of using this new pipeline, such as the ability to more thoroughly explore the design space of semantic interaction or enable rapid prototyping, as well as the limitations.

Specifically, we note the following contributions:

1. A review of the necessary components to accomplish semantic interactions in visual analytics tools (i.e., model composability, inverse computations, pipeline bidirectionality);
2. A new conceptual pipeline that incorporates the necessary components of semantic interaction;
3. A set of examples demonstrating how this pipeline is capable of capturing semantic interaction designs in both existing and new visual analytics tools.

2 RELATED WORK

To fully capture the complexity of semantic interactions, our goal in this work is to define a new conceptual pipeline that depicts the structure of the feedback loop between the various data processing components of the semantic interaction-enabled visual analytics pipeline. We justify the need for such a pipeline by surveying the current state of commonly-referenced pipeline models in information visualization and visual analytics as well as exploring the breadth of pipelines used to model existing visual analytics tools.

2.1 Existing Pipelines for Visual Analytics Processes

The fields of information visualization and visual analytics rely on computational and visual pipelines to convert data into visual displays. For example, Figure 1[2] (top) shows the Sensemaking Loop defined by Pirolli et al. [23], which identifies the different mental processes involved in transforming raw data into a presentation of a formalized hypothesis. Located below the Sensemaking Loop in this figure are both the information visualization pipeline of Card et al. [6] and the visual analytics task process of Keim et al. [16].

These pipelines model a high-level representation of how raw data is transformed to a final visualization or presentation and are

*e-mail: dowlingm@vt.edu
†e-mail: jw87@vt.edu
‡e-mail: phauck@vt.edu
§e-mail: adamq@vt.edu
¶e-mail: npolys@vt.edu
‖e-mail: north@cs.vt.edu

[1] In this work, we employ the convention of referring to individual data items as *observations* and the properties of those observations as *attributes*.

[2] Larger versions of all figures are provided in the supplementary material.

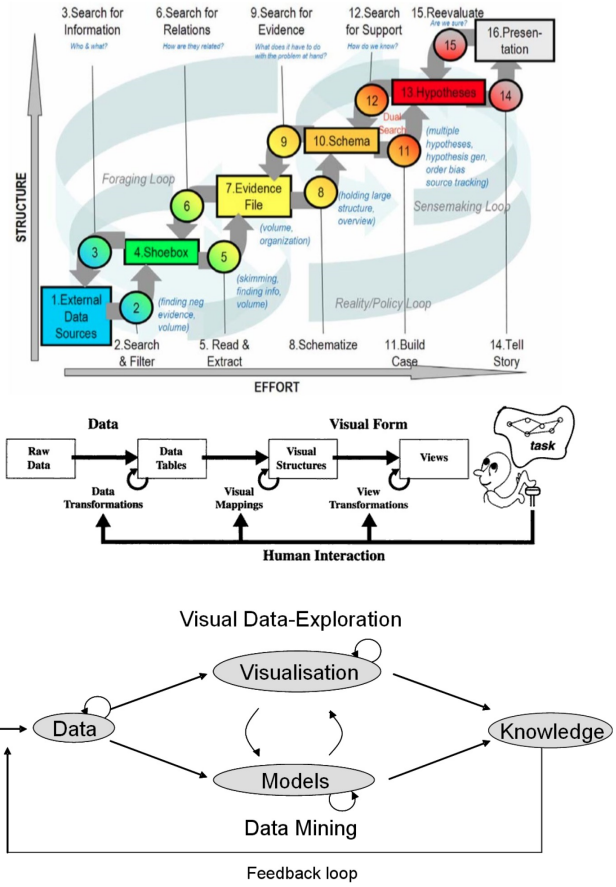

Figure 1: **(top)** In the Sensemaking Loop [23], sensemaking contains a backward (or inverse) process for each forward step. Chaining these combined forward/inverse processes as composable processes yields a full bidirectional cognitive pipeline. **(middle)** The information visualization pipeline presented by Card et al. [6] does not specifically model semantic interactions. **(bottom)** The visual analytics model provided by Keim et al. [16] provides a high-level overview of the structure of visual analytics knowledge discovery, but lacks detail in defining how mathematical models are used to interpret semantic interactions. In order to support semantic interaction, a different pipeline structure is necessary.

quite generalizable, but the resulting trade-off is that these pipelines abstract any details of the mathematical model(s) and visualization(s) into single nodes in the graph. For example, the emphasis on a high-level abstraction on visual analytics task processes means that the pipeline defined by Keim et al. does not explicitly discuss interaction. Similarly, the focus on the mental processes in the Sensemaking Loop means that mathematical models are not considered in this pipeline. In contrast, the interactions described in the pipeline from Card et al. represent methods to directly manipulate parameters of mathematical models, such as slider interactions. While this is interaction, we do not define this to be *semantic* interaction as no interpretation is necessary by any model for this interaction; the value provided by the analyst is simply stored and used. Thus, the precise mechanisms used to process and visualize the data or to interpret semantic interactions are not adequately captured in either of these pipelines. Looking at other pipeline representations, such as those presented in a survey of analytical pipelines by Wang et al. [29], we find the same limitations for semantic interaction tool design. Thus, these pipelines are insufficient for capturing how to

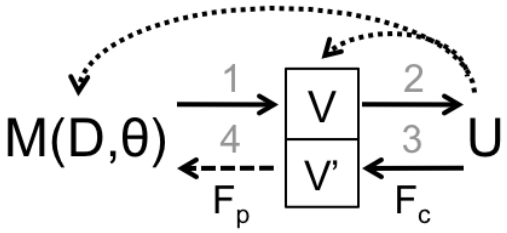

Figure 2: V2PI [19] is a mathematical representation of semantic interaction. This framework supports the creation of a visualization V. When the analyst U manipulates V to form V' via a semantic interaction, this triggers a manipulation of the parameters θ that influence model M. The parameterized feedback (F_p) represents an inverse process similar to what is described by the Sensemaking Loop, in which the interaction is interpreted as a set of updates to model parameters.

support semantic interaction in visual analytics tools.

In contrast to these pipelines, V2PI [19] provides a statistical semi-supervised machine-learning methodology for realizing semantic interaction. The V2PI pipeline (Figure 2) supports interactivity for visualizations and relies on both proven statistical methods and the analyst's judgment. In this pipeline, a visualization is created by processing data and parameters through a mathematical model. This visualization is presented to the analyst to evaluate. The analyst can directly manipulate the visualization, referred to as cognitive feedback. This cognitive feedback is translated into parameterized feedback, typically via machine learning, which updates the model through newly learned parameter values. As a result, a new visualization is created based on the analyst's interaction. Given this definition of V2PI, we assert that V2PI appropriately captures the basics of semantic interaction. However, V2PI only permits the exploration of a single mathematical model to accomplish such interactions. Thus, while V2PI may be able to capture simple visual analytics tools which contain a single mathematical model, it is not capable of representing tools with multiple models, such as StarSPIRE [4].

2.2 Tool-Specific Pipelines

Semantic interaction tools have become increasingly varied in interaction methods and purpose. Here, we consider tools as implementing semantic interaction if they meet three characteristics: (1) analysts can directly manipulate visualization, (2) the tool interprets the analyst's intent from these interactions to update some learned model (typically a semi-supervised machine learning algorithm), and (3) the goal of these models and interactions is to enhance the cognitive Sensemaking Loop [23].

To incorporate semantic interaction, some tools leverage the V2PI framework previously discussed, including ForceSPIRE [12], StarSPIRE [4], and Andromeda [25]. In each of these tools, the analyst directly interacts with observations in a dimension-reduced projection of data. These interactions drive a model that learns the relative importance of the attributes in the high-dimensional data space.

Additional tools also support interacting with a projection, but were not explicitly created with the V2PI framework in mind. Examples include the LAMP framework described by Joia et al. [15] and the extention to iLAMP [7], the technique described by Mamani et al. [20], Dis-Function [5], the technique defined by Paulovich et al. [22], and the tool developed by Molchanov et al. [21]. However, semantic interaction can extend beyond interactions with projected observations to learn attribute weights. For example, both InterAxis [17] and AxiSketcher [18] use interactions on observations in the projection to update the axes of the projection. Intent Radar [24]

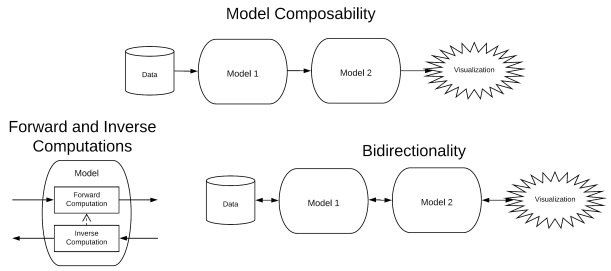

Figure 3: A representation of our three characteristics for a new semantic interaction pipeline: Model Composability, Bidirectionality, and Model Inversion. Model Composability refers to how different mathematical models must work together to produce the desired visualization. Bidirectionality allows interactions to drive updates to the underlying models. Model Inversion refers to the pairs of a forward computation with an inverse computation. The inverse computation supports the translation of semantic interactions into manipulations of model parameters.

introduces interactive intent modeling, allowing an analyst to provide feedback by dragging or clicking keywords, increasing the relevance by moving the keyword closer to the center or decreasing it by moving it outward in the radar interface. Moving away from projection-based tools entirely, Podium [28] is a tabular ranking tool in which an analyst reorders rows (still observations) in the table while the tool learns the attributes important to the current ranking scheme. iCluster [9] provides analysts with the ability to interactively move documents into clusters, learning the attributes important to the current clustering scheme. Similarly, ReGroup [1] interactively learns a model of group membership as an analyst adds members to groups.

Although each of these tools employ semantic interaction, few of the papers offer associated pipelines to properly capture the complexity involved with the interaction. Moreover, the pipeline representations are diverse, ranging from high-level abstractions to more detail-oriented representations. In Figure 5, we show a subset of these pipeline representations, including Andromeda [26], StarSPIRE [4], Dis-Function [5], Piecewise Laplacian Projection [22], and the pipeline provided by Mamani et al. [20] to describe their technique. Although the pipelines for Andromeda and StarSPIRE arguably achieve the highest level of detail to capture the semantic interaction therein, we feel that these pipelines can be improved to focus even more on the mathematical models used to create the visualization and interpret the semantic interactions therein (which is further explained in Section 5). In contrast, the pipelines for Dis-Function, Piecewise Laplacian Projection, and the technique by Mamani et al. are high-level pipelines which focus on the general concepts behind how the associated tools and techniques work. The trade-off in these, just as with the pipelines by Card et al. and Keim et al., is that the mathematical models used are abstracted away, making it difficult to determine how the semantic interactions therein are accomplished.

3 Characteristics of Semantic Interaction in Visual Analytics Tools

When comparing the characteristics of the visual analytics tools discussed in the previous section, we note that there are several commonalities. Combined with the ideas from the Sensemaking Loop [23], we define three properties as necessary for supporting semantic interaction in visual analytics tools. Each of these properties map directly to structures required to represent the complexity involved in modeling semantic interactions in a generalized pipeline for visual analytics tools.

3.1 Model Composability

The first characteristic we identified is that each mathematical model used to process the data as it works its way to the final visualization has specific input and output requirements. This hints to how these mathematical models must be composed to work together within the pipeline in order to produce the desired visualization. For example, PCA requires numerical high-dimensional data as input to produce low-dimensional coordinates as output. Therefore, any models preceding PCA must produce these high-dimensional data, and any models after PCA must be able to work with the low-dimensional coordinates as input. As another example, Weighted Multidimensional Scaling (WMDS) accepts numerical high-dimensional data as well as a set of attribute weights as input to produce low-dimensional coordinates as output. Thus, while the output is the same as with PCA, the input requirements have changed. This change must be accounted for in either data preprocessing steps or in a mathematical model that precedes the WMDS model. Therefore, model composability is a fundamental characteristic of semantic interaction and is represented by the top row of Figure 3.

3.2 Forward and Inverse Computations

While the model composability characteristic may seem simple or intuitive, it has important implications for the structure of a pipeline that captures semantic interaction. For example, Andromeda [26] uses WMDS (Weighted Multi-Dimensional Scaling) to produce low-dimensional coordinates given a set of attribute weights. However, observation-level interaction, or OLI (which is a type of semantic interaction), expands the WMDS model by providing new low-dimensional coordinates from which to *learn* a new set of attribute weights. Given that the dataset is treated as a constant, this effectively inverts the WMDS computation.

We find this type of computation inversion common in tools with semantic interaction [4, 5, 8, 20, 22, 26, 31]; it is this inversion which defines the learning or interpretation necessary to realize semantic interaction. Therefore, we propose that computation inversion is a required characteristic for visual analytics tools that support semantic interaction. Thus, our new pipeline must capture both forward and inverse computations for a given model. This concept is represented by the bottom right of Figure 3. Combined with the aforementioned model composability, this means that each mathematical model must fulfill composability requirements for its inverse computation as well as its forward computation.

3.3 Looping Sensemaking via Bidirectionality

Taking the model composability and forward and inverse requirements a step further begins to imply a required bidirectionality in how the models are used together. In other words, each model must fulfill composability requirements for both its forward and inverse computations. Combine this with the fact that the forward computations help produce the given visualization and the inverse computations help interpret an interaction, then the pipeline must be bidirectional to support a looping structure. This bidirectional structure can be seen in both StarSPIRE [4] and Andromeda [26], which each use inverse computations of their models to interpret semantic interactions, followed by the forward computations to generate updated visualizations.

Referring back to the Sensemaking Loop [23], we see a similar structure between pairs of processes that allow for information to be progressively transformed. These pairs of processes allow the transformation to occur in both forward and inverse directions, implying that there is a concept of looping between these collections of information. Thus, bidirectionality in a pipeline to represent semantic interaction mimics this natural process of incrementally building information to generate an output and then reassessing and refining information to produce a better output. This approach captures the

concept of *incremental formalism* [2, 3, 27] in the cognitive sensemaking processes, in which analysts incrementally improve their mental models of the data through interaction, and represents that cognitive process formally as a machine learning process.

However, the Sensemaking Loop as well as existing semantic interaction tools [8, 20, 22, 26, 31] also indicate that it is not always necessary to iterate through the entire pipeline and all models to generate the desired results. As an example, Andromeda [26] uses the aformentioned semantic interaction of OLI. When this occurs, an inverse computation is triggered that determines new attribute weights given a set of low-dimensional coordinates. However, since all the observations are already visualized, there is no need to pull any additional data into the pipeline. Thus, there is no need for any new data processing, meaning processing can skip to immediately recalculating new low-dimensional coordinates for all observations using the learned attribute weights. In the Sensemaking Loop, a similar concept is represented by the fact that the analyst does not have to go all the way back to the external data sources every time he/she wishes to refine information. For instance, an analyst refining an evidence file may only need to reread or perhaps read more of a file that has already been accessed rather than foraging for a completely new file.

These examples reveal an important feature with respect to this bidirectionality characteristic: the ability to short circuit the rest of the pipeline when appropriate. This is a key new feature of a multi-model pipeline not found in earlier definitions [4]. Short circuiting happens when the inverse computation of a model does not need to send the interaction any further down the pipeline. Thus, instead of running the entire pipeline, we can short circuit to skip over unnecessary components of the pipeline, executing the forward computations beginning with the last model used to perform an inverse computation. From there, other models that were also updated should also have their forward computations rerun to produce an updated visualization. While the bidirectionality chatacteristic is represented in the lower-left of Figure 3, this short circuiting concept is depicted by the upward arrow between the inverse computation and the forward computation in the lower-right of Figure 3.

4 Components of a Semantic Interaction Pipeline for Visual Analytics Tools

4.1 A New Semantic Interaction Pipeline

When evaluating traditional visual analytics models (e.g., Keim et al. [16]), we note that there is rarely a distinction between different models that may be used in the pipeline. Thus, model composability is not well-represented in these existing models. Furthermore, while bidirectionality may be represented on some level, the manner in which the visual analytics pipeline handles this bidirectionality is not discussed or represented in detail. Additionally, there is no representation of inverse computations within the models. Therefore, there is a need for a new pipeline for visual analytics tools that better captures these characteristics of semantic interaction.

For our proposed new pipeline, we require properties of the pipeline to map back to the model composability, model inversion, and bidirectionality characteristics discussed previously. To capture these characteristics, we define this new pipeline to consist of three components, which are further described in the following subsections: a Data Controller, a set of Models, and the Visualization[3]. This new pipeline is shown in Figure 4. The forward and inverse computation characteristic is addressed by having each Model represent a set of such computations. Arrows between Models and other pipeline components indicate how the input and outputs requirements for each component line up[4], thereby addressing the model composability characteristic. The bidirectionality of the overall pipeline is handled through transitions between these computations, using the forward computations in the projection direction and the inverse computations in the interaction direction to loop through the pipeline in response to a semantic interaction. Upward arrows between inverse computations and forward computations of a given Model show when the pipeline short-circuits rather than iterating through the entire pipeline to interpret a semantic interaction. Thus, this proposed structure accurately captures the power and complexity of semantic interactions.

4.2 Models

As evident by our discussion thus far, the primary focal point of our proposed pipeline is the Models. This is because the Models alone must encompass two of the three identified characteristics of semantic interaction: model composability and model inversion. To capture the inversion characteristic, each Model consists of a set of computations: a forward computation that is used to help produce the desired Visualization and at least one inverse computation that is used to help interpret an interaction by updating the inputs to the forward computation. These inverse computations can come in many forms, including precise mathematical inverses [25], heuristic inverses [31], and probabilistic inverses [14].

The forward and inverse computations of the Model naturally have input and output requirements, hinting at the given Model's composability with other pipeline components, whether they be other Models in the pipeline, the Data Controller, or the Visualization itself. These input and output requirements and how they are addressed is implied by how the Model connects to these other components in the pipeline. In Figure 4, this connectivity between a given Model and other pipeline components is represented by the arrows between these pipeline components. These arrows therefore represent the process used to both create the desired Visualization and interpret interactions within the Visualization. However, it is important to note that while these arrows provide an overview of how each Model is composed within the pipeline, the specific details of how composability requirements are met are left to the corresponding text accompanying the pipeline. To help provide more details for these composability requirements through the pipeline itself, the pipeline can be further annotated. For example, the arrows throughout the pipeline can be annotated with mathematical variables used to represent the inputs and outputs of each pipeline component. The trade-off in doing so is that such annotations may lead to visual clutter or initial confusion as to what these annotations mean.

Since these arrows represent the processes of Visualization production and interaction interpretation, we begin to note how the bidirectionality requirement is also addressed through this new pipeline. That is, there are a set of arrows that flow through each Model in the pipeline in a forward direction to produce the given Visualization as well as a set of arrows that flow in a backwards direction to interpret interactions within the Visualization. Thus, the manner in which the Models are represented in the pipeline alongside the other pipeline components denotes the pipeline's bidirectionality, thereby capturing this final characteristic of semantic interaction.

However, there is an additional nuance of bidirectionality that is also captured within each Model: being able to short-circuit the pipeline when no further computation is needed to interpret the given interaction. This is represented by an arrow between the inverse computation of a Model and its forward computation. Referring to our previous example with Andromeda, OLI does not need to communicate with any other pipeline component. Therefore, there is no need to send this interaction further down the pipeline, allowing the Model to short-ciruit. This immediately triggers a recalculation

[3] To differentiate common terms (e.g., a mathematical model) from pipeline components, we capitalize pipeline components (e.g., Model).

[4] As shown in the SIRIUS pipeline in Section 6.2, it is certainly possible to create non-linear pipelines that model how subsets of models collaborate to handle different groups of semantic interactions.

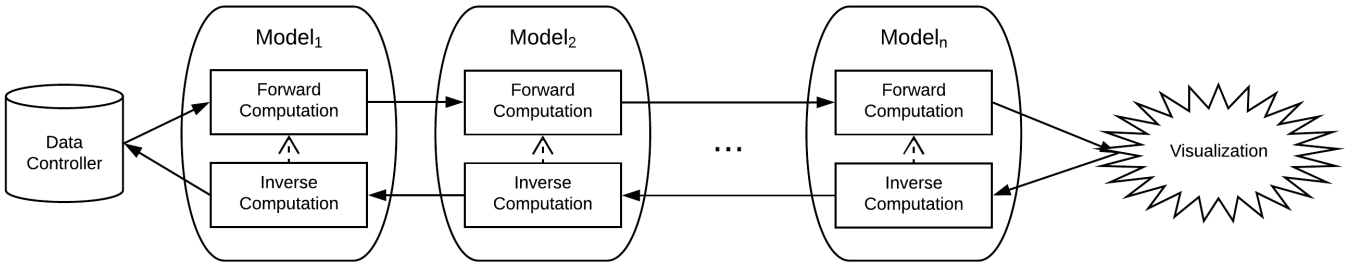

Figure 4: Our new pipeline for semantic interaction in visual analytics tools, created from the combination of the three characteristics shown in Figure 3. Model composability is shown through the chaining of a series of models horizontally in the pipeline. Bidirectionality results from the separated forward (top) and inverse (bottom) paths through the models. Model inversion is shown through the pairing of a forward computation and an inverse computation in each of the models. This representation also shows short circuiting arrows that connect the inverse and forward computations in the Models. The resulting structure captures how data is transformed into a Visualization and how semantic interactions are interpreted to update the parameters of the forward computations of the different Models.

of the low-dimensional coordinates of the data using the newly calculated attribute weights, enabling the Visualization to update as soon as possible.

4.3 Data Controller

Which Models are supported in a pipeline is highly dependent on the data being used. Therefore, to better contextualize the Models in the pipeline, our pipeline necessitates a Data Controller to serve as the main access point to the underlying data that is being visualized. Its key purpose is to retrieve the raw data and any possible metadata as well as to transform this data into a form usable to the Models through data preprocessing. Thus, the Data Controller can enable analysts to view the raw data directly or allow the pipeline to pull additional data to process and visualize. This means that a Data Controller is specific to a particular type of data or dataset.

4.4 Visualization

Finally, it is difficult to understand or appreciate a Model without understanding the Visualization being used and the interactions enabled therein. Therefore, our new pipeline also requires a Visualization component. Firstly, the Visualization must define how the output from the Models are mapped to different visual elements in the Visualization. Additionally, this pipeline component determines how the visual elements are interacted with and which Model(s) should be used to interpret this interaciton. Thus, an interaction within the Visualization initiates inverse computations in the Models of the pipeline to interpret the given interaction and produce an updated Visualization.

5 Using the Pipeline for Existing VA Tools

With this pipeline structure, we have the ability to well describe the complexity of semantic interaction in existing visual analytics tools. To exemplify this, we focus on the five of the visual analytics tools and techniques we described in Section 2.2. Figure 5 shows a side-by-side comparison of the pipelines provided in the perspective papers for each tool or technique and how to represent each using our newly proposed pipeline. From A to E in Figure 5:

Andromeda [26] provides a scatterplot projection of numerical high-dimensional data using Weighted Multidimensional Scaling (WMDS). In this projection, the analyst can perform a semantic interaction called Observation-Level Interaction (or OLI) in which he/she provides new low-dimensional coordinates for a subset of the observations. From these observations, new attribute weights are learned, which are then used to update the low-dimensional coordinates of all the observations. Both these interactions manipulate the parameters for the WMDS mathematical model. Therefore, three pipeline components are needed to represent Andromeda using our new pipeline: a CSV Data Controller, a WMDS Model, and the Visualization. The CSV Data Controller reads in a specified CSV file of numerical high-dimensional data and normalizes each attribute using z-scores. It also initializes each attribute weight to be $1/p$, where p is the number of attributes in the dataset. Using this normalized data and attribute weights, the forward computation of the WMDS Model determines the low-dimensional coordinates for each observation. The Visualization component displays the low-dimensional coordinates and the attribute weight values. The inverse computation then determines new attribute weights based on the analyst-defined low-dimensional coordinates. At this point, the pipeline always short-circuits to run the forward computation and determine (and then display in the Visualization) new low-dimensional coordinates for all observations; the CSV Data Controller is never needed beyond the data preprocessing steps since all observations are always displayed, meaning no further computation from the Data Controller is needed.

StarSPIRE [4] provides a scatterplot-like projection for queried text data. Thus, the analyst must perform a query and perform subsequent queries or interactions in order to pull documents into the visualization. To represent this process in our new pipeline, four pipeline components are needed: a Text Data Controller, a Relevance Model, a Force-Directed Model, and a Visualization. The Text Data Controller initializes a set of extracted entities from the document set and ensures each document has an associated TF-IDF value for every entity. After providing references to the locations of the documents themselves and an initialized set of entity weights, $1/p$, the forward computation of the Relevance Model computes the relevance of all documents according to the current entity weights. Only the top n documents above a given threshold will be added to the visualization. Thus, the Relevance Model acts as a query filter for which documents are passed to the Force-Directed Model. The forward computation of the Force-Directed Model determines the low-dimensional coordinates of each document passed to it, using the same entity weights to place similar documents near each other. The Visualization then uses both the low-dimensional coordinates and the relevance (mapped to node sizes) to display the documents. Semantic interactions in StarSPIRE can cause the Force-Directed Model and the Relevance Model to learn new entity weights in their inverse computations. Thus, when the analyst manipulates the document positions or relevances, new entity weights representing the analyst's interest are learned and then used in the forward computation to update the visualization accordingly.

Dis-Function [5] displays, among other views, a scatterplot of projected pairwise distances using Principal Component Analysis (PCA). An analyst is able to perform semantic interactions by providing new low-dimensional coordinates for observations, causing the tool to learn new attribute weights for PCA and reprojecting the observations using these new weights. This behavior is quite similar

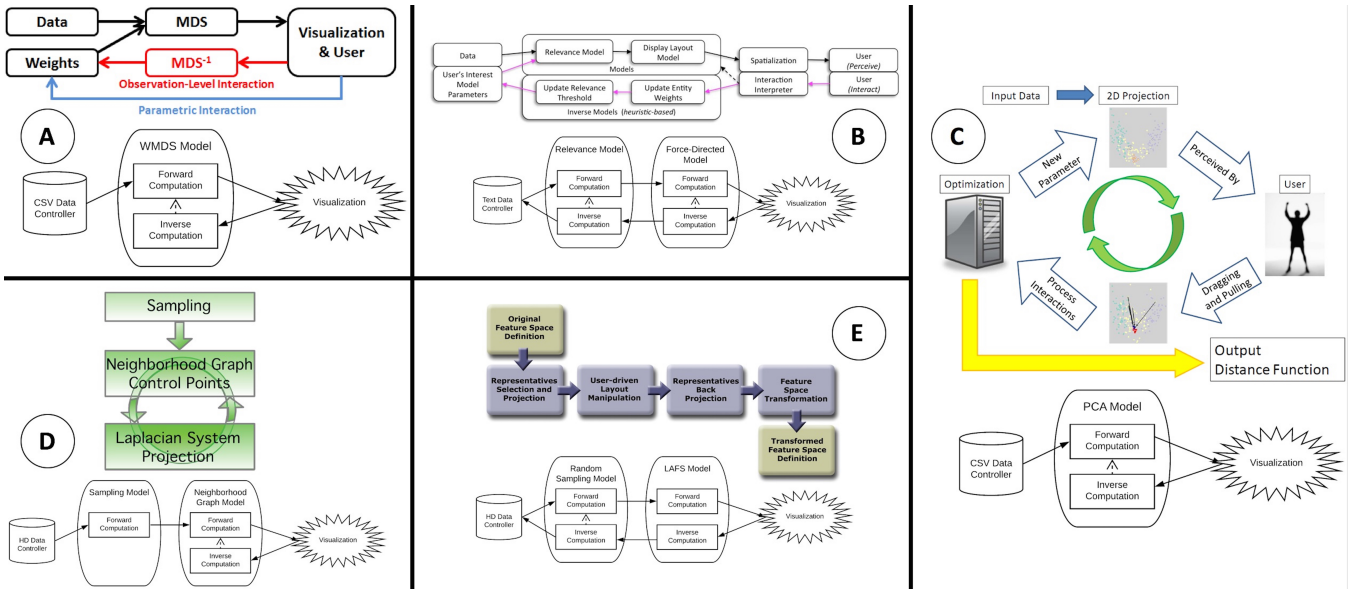

Figure 5: Using the proposed semantic interaction pipeline shown in Figure 4, we can now model the behavior of existing semantic interaction tools like (A) Andromeda [26], (B) StarSPIRE [4], and (C) Dis-Function [5], (D) Piecewise Laplacian Projection [22], and (E) Mamani et al. [20]).

to Andromeda, thereby using a PCA Model in place of Andromeda's WMDS Model in its pipeline.

Paulovich et al. [22] present a Piecewise Laplacian projection tool in which samples are drawn from a full dataset, control points are created for each sample, and a neighborhood graph is constructed for the full dataset. As an analyst manipulates the projection through semantic interactions, these control points and neighborhood graphs dynamically update. Using our new pipeline, we can model this process using a Sampling Model to perform the sampling step and a Neighborhood Graph Model to perform the projection. In the Sampling Model, the forward computation performs the initial sampling step, which then feeds into the forward computation of the Neighborhood Graph Model to specify the control points and project the data. Semantic interactions can be performed by directly manipulating the projection, causing the Neighborhood Graph Model to learn a new projection through its inverse computation and short-circuiting to rerun its forward computation. Since there is no semantic interaction defined that alters the Sampling Model, this model effectively has no inverse computation defined, highlighting the possibility for additional semantic interactions to be included in this type of tool.

Mamani et al. [20] propose a tool similar to that of Paulovich et al., though the projection is based on local affine mappings rather than Laplacian. Still, the basic process of sample first and project second remains the same in the forward computations. This means that the pipeline representation of this tool uses a Local Affine Force Scheme Model in place of the Neighborhood Graph Model described above. However, the process of responding to semantic interaction also incorporates the inclusion of a new set of samples, thereby incorporating an inverse computation in both the Local Affine Force Scheme Model and the Random Sampling Model.

6 Using the Pipeline for New Visual Analytics Tools

In this section, we illustrate three visual analytics prototypes that have been developed using our new visual analytics pipeline to further explore the design space for semantic interaction. These prototypes handle different types of data (numerical and text) and alter similar Models to create distinct Visualizations and semantic interactions therein. Each of the prototypes are discussed in the following format:

- **Motivation:** We begin by motivating the creation of the prototype, describing why such a tool is useful and what we could learn from it.
- **Visualization and Semantic Interactions:** We describe the Visualization developed and the semantic interactions enabled therein to provide context for the various pipeline components.
- **Pipeline:** We discuss how the given Visualization and semantic interactions are accomplished mathematically through our new visual analytics pipeline. Since we define the Visualization previously, we effectively separate the discussion of this pipeline component from the others to improve clarity.

6.1 Cosmos

6.1.1 Motivation

The Cosmos pipeline was created to explore how to incorporate the Relevance Model from StarSPIRE [4] with a stricter notion of similarity than a force-directed layout, such as is accomplished in Andromeda's WMDS Model [26]. With these two models, analysts can query for specific terms or documents in the dataset, view the raw text from a document, manipulate a document's relevance, and directly manipulate the projection of the documents.

6.1.2 Visualization and Semantic Interactions

As shown at the bottom of Figure 6, the Cosmos Visualization consists of two panels. While the left panel is an interactive WMDS projection of the documents, the right panel displays the details for a single selected document. Unlike in Andromeda, the WMDS projection is initially empty, requiring the analyst to perform a search to bring documents into the Visualization. After documents are placed on the screen, their relevance calculations are mapped to the sizes of the projected observations. The analyst can then use an array of interactions to manipulate the Visualization. For example, double-clicking an observation populates the panel to the right of this projection with information specific to the corresponding document. This includes an interactive relevance slider, the label of the projected observation, and the raw text of a document and associated notes. The analyst also has the ability to remove a document from the Visualization by clicking a button on this panel.

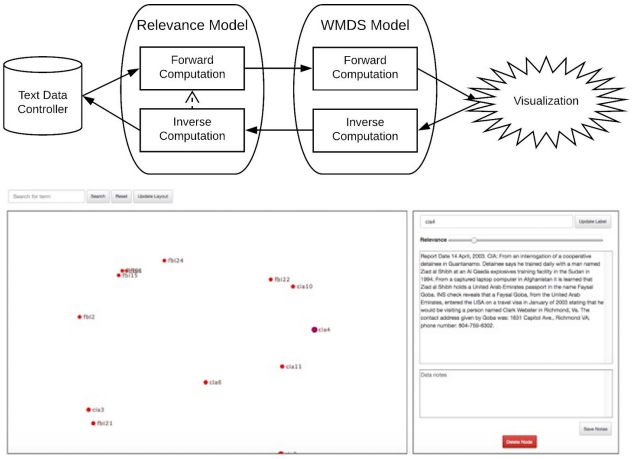

Figure 6: **(top)** Our pipeline representation of Cosmos consists of a Text Data Controller, Relevance Model, WMDS Model, and a Visualization. The Relevance and Similarity models each handle a different component of manipulating the data to create the Visualization. **(bottom)** The Cosmos interface allows analysts to interact with documents, manipulating their similarity and relevance throughout the exploration of the dataset.

As in Andromeda, the analyst can perform the semantic interaction of OLI in Cosmos by clicking and dragging documents of interest in specific locations (to denote their desired similarity/dissimilarity) and clicking an "Update Layout" button. This triggers a recalcualtion of attribute weights using only the low-dimensional observations the analyst interacted with. However, Andromeda stops there and reprojects the entire dataset to create a new Visualization; Cosmos continues its interpretation of this interaction by automatically performing a query for more documents on behalf of the analyst, guided by these new attribute weights. After combining the new documents with the old documents, the relevance of each document is recalculated, and the data is reprojected to genereate a new Visualization.

In addition to OLI, Cosmos affords an additional semantic interactions through its "Relevance" slider. This slider is available for a selected document, as seen in the details panel at the bottom of Figure 6. When this slider is manipulated by the analyst, the new attribute weights are calculated which best estimate the analyst-defined relevance for the given document. If the relevance for the document is increased, then this interaction is interpreted as reflecting a document that the analysts likes and would want to see more of. Therefore, this interaction also triggers an automatic query for more documents, which uses these new attribute weights. Otherwise, the pipeline simply continues its process by recalculating all document relevancies and reprojecting all documents to create a new Visualization.

6.1.3 Pipeline

The Cosmos pipeline is shown at the top of Figure 6. Note that this pipeline is similar to the StarSPIRE pipeline. In addition to the replacing the Display Similarity Model with the WMDS Model, we have modified the Data Controller and Visualization as well. Each component of this pipeline is described below:

Text Data Controller: For this pipeline, we modified the Data Controller from Andromeda to work with text documents. To do so, we first assume that the uploaded CSV file contains the TF-IDF values for entities extracted from the document set. This assumption allows us to skip additional preprocessing steps to focus instead on the Models themselves and their influence on the Visualization.

Once uploaded, this Text Data Controller reads the data from the CSV file and preprocesses the data in the same manner as Andromeda's Data Controller to ensure each entity is treated equally by the Models. This is accomplished by normalizing each entity's TF-IDF values to be within a standard deviation of 1. Additionally, this Text Data Controller adds references to the flat files for each document, which are assumed to be in a single directory.

The final role of the Text Data Controller is to initialize a set of entity weights for the Relevance Model and Similarity Model to use, thus fulfilling the composability requirements for the forward computations in these Models. We initialize these weights to be $1/p$, where p is the number of extracted entities in the uploaded CSV file. These weights, along with the other document-related data, are sent along the pipeline to the Models.

Relevance Model: We drew inspiration from StarSPIRE [4] to create our Relevance Model. The Relevance Model uses the same set of attribute weights that the WMDS Model does (which is described next), but in a different manner. In the forward computation, this model computes the relevance of a document given a set of attribute weights as a linear combination of those weights and the document's TF-IDF values. This simple relevance calculation combined with a threshold determines which documents are passed on to the WMDS Model. That is, the Relevance Model acts as a filter that determines which documents are visualized. The forward computation has a matching inverse computation to calculate the entity weights that produce a relevance value for a given document.

Additionally, the Relevance Model is responsible for querying for new documents to display, whether the query was initiated by the analyst by searching for a term or automatically by the Cosmos itself (e.g., on OLI or increasing a document's relevance). Using the entity weights, the Relevance Model finds the top n most relevant documents that are above the relevance threshold. This ensures that only highly relevant documents are displayed while also guaranteeing that the analyst will not be overwhelmed by too many documents appearing in the Visualization at once. If querying is not necessary to interpret the given interaction (e.g., when the relevance value for a document is decreased), then the Relevance Model simply short circuits, allowing for immediate recalculation of the relevances of all documents currently being displayed.

WMDS Model: The role of the WMDS Model is to spatialize documents according to their similarity based on a given set of attribute weights, just as is accomplished in Andromeda [26]. However, Cosmos relies on data passed from the Relevance Model to define which documents should be used as well as the entity weights. The forward computation uses these weights to project the high-dimensional data in the Visualization.

The WMDS Model also uses the same inverse computation defined by Andromeda, enabling OLI interactions. This calculates the entity weights based on low-dimensional coordinates of documents in the Visualization. However, Cosmos also performs an automatic query based on these new entity weights. Since this automatic query always occurs after OLI and because the Relevance Model is responsible for such querying, the WMDS Model never short-circuits. After the Relevance Model also recalculates document relevances, the WMDS forward computation is then run to determine new low-dimensional coordinates for all documents to be displayed in the Visualization.

6.2 SIRIUS

6.2.1 Motivation

In the primary SIRIUS paper [8], we noted that analysts often think about the observations and attributes in similar manners. In other words, there is a symmetry between how analysts analyze attributes and observations of a dataset. Therefore, there is a need to develop visual analytics tools that afford this symmetric thought process, leading us to develop SIRIUS (Symmetric Interactive Representations In a Unified System). While further details of SIRIUS are

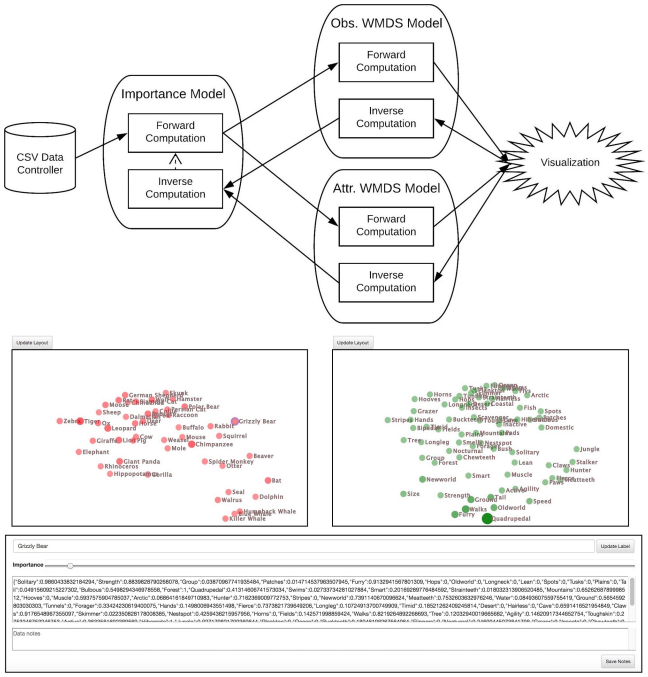

Figure 7: **(top)** Our pipeline representation of how SIRIUS produces the observation and attribute WMDS projections and how this tool interprets semantic interactions therein using our new proposed pipeline. This is accomplished using a CSV Data Controller, Importance Model, two WMDS Models, and a Visualization. **(bottom)** This Visualization consists of two interconnected, interactive WMDS projections: one for the observations and one for the attributes of a high-dimensional dataset.

described in [8], we focus on its pipeline representation here.

6.2.2 Visualization and Semantic Interactions

The SIRIUS Visualization in Figure 7 consists of two main panels: a left panel for a projection of the observations and a right panel for the projection of the attributes. Both projections are WMDS projections, with node sizes and opacities reflecting the importance of the given observation or attribute. Both of these panels enable the same semantic interaction of OLI previously described. However, instead of only updating one projection, this semantic interaction updates both projections in SIRIUS.

Below these two main panels is a third panel that provides an interactive "Importance" slider that allows the analyst to define the importance of a selected observation or attribute. The associated raw data is also provided in the text field in this panel for the analyst's convenience. Manipulation of this "Importance" slider is a semantic interaction that triggers a recalculation of attribute weights and observation weights, thereby resulting in updates to both projections in SIRIUS.

6.2.3 Pipeline

CSV Data Controller: The Data Controller used in SIRIUS is virtually the same as the one used in Andromeda. The main difference is that the Data Controller in SIRIUS must normalize both the original data as well as its transpose separately. This enables the projections to represent all observations and attributes without an artificial emphasis placed on any one attribute or observation due to naturally higher values (e.g., height vs weight of a person).

Importance Model: We again drew inspiration from StarSPIRE's Relevance Model as it provides a simple method for the forward computation to calculate the importance (i.e., relevance) for any one observation or attribute using a linear combination of attribute or observation weights and the associated data for the given observation or attribute (respectively). However, these calculations also make it easy to translate the importance of attributes to the importance of observations and vice versa by expanding the importance calculation for a single observation or attribute to calculate the importance of all observations or all attributes at once. Thus, these importance calculations enable a recalculation of observation weights based on entity weights and vice versa. Since the WMDS Models (discussed next) use the same set of weights, this means that both projections update based on a single interaction in either projection.

To enable semantic interactions, the Importance Model's inverse computations begin when the analyst manipulates the "Importance" slider. This triggers an inverse calculation of the weights that produce the analyst-defined "importance" value using equations similar to those used in the Relevance Model's inverse computation from Cosmos. For example, if the analyst manipulates the "importance" value for an attribute, then the observation weights to produce that "importance" value are calculated using one of these inverse computations. However, these new weights are then used to determine new attribute weights. To enable more insights for attribute similarities/correlations, these attribute weights from the inverse computation are then used to recalculate new observation weights in the forward computation. These final sets of weights are then used in the WMDS Models to reproject the data in the Visualization.

The Importance Model performs a similar set of calculations on OLI. For example, if OLI is performed in the observation panel, then a new set of attribute weights are determined in the WMDS Model. However, to translate this change to changes in the attribute projection as well, new observation weights must be determined. As a result, both new sets of weights are passed to the WMDS Models to determine the new positions of the nodes in both panels.

It is important to note that just as with Andromeda, SIRIUS assumes that all observations and attributes are used from the beginning. Since no querying for new data is performed, SIRIUS never needs to communicate with the CSV Data Controller again, causing the Importance Model to always short-circuit.

WMDS Models: As seen in Figure 7, SIRIUS consists of two WMDS Models: one for the projection of observations and one for the projection of attributes. The Observation WMDS Model uses the normalized form of the original dataset and the same attribute weights used by the Relevance Model to determine the low-dimensional coordinates for each observation using the same WMDS equation from Andromeda and Cosmos. Similarly, the Attribute WMDS Model uses the normalized form of the transposed dataset and the same observation weights used by the Relevance Model to determine the low-dimensional coordinates for each attribute.

Each of these WMDS Models enable OLI separately. That is, OLI can only be performed on one panel at a time. Then, an inverse WMDS computation similar to the computation described in [26] is used to calculate a new set of weights. These weights are then passed to the Relevance Model to enable updates in both panels.

6.3 A Cluster-Based Visualization

6.3.1 Motivation

We have also begun investigating how the introduction of explicit clustering assignments affect the ways in which analysts perceive and interact with projections [31]. The technique itself can make use of a variety of layout and clustering techniques, but the following implementation describes an instantiation using a force-directed layout for similarity projection and k-means clustering on the projection to automatically group similar observations. Analysts directly interact with the projection using semantic interactions to alter clustering assignments of the observations in order to manipulate the underlying mathematical models.

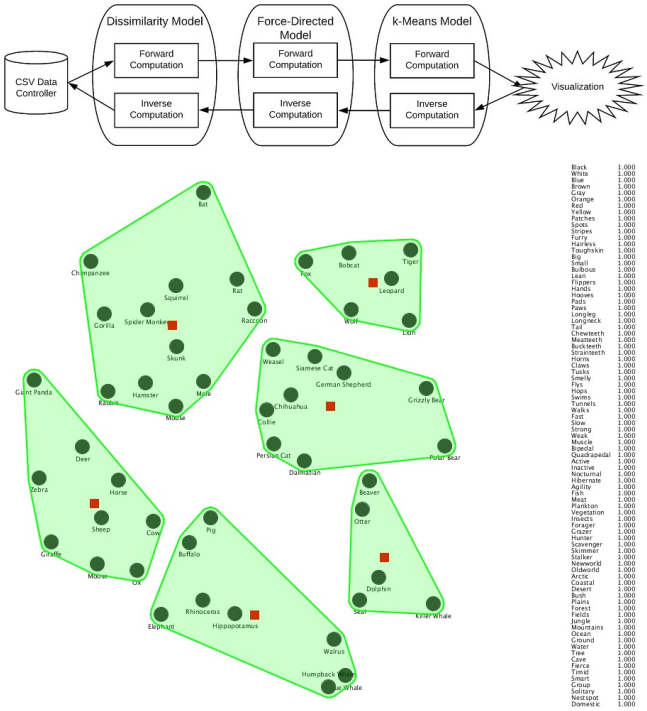

Figure 8: **(top)** Our pipeline representation for how the cluster-based visualization by Wenskovitch and North is created and semantic interactions therein are interpreted. This is accomplished using a CSV Data Controller, Dissimilarity Model, Force-Directed Model, k-Means Model, and Visualization. **(bottom)** The clustering interface allows analysts to explore related groups of observations depending on the learned attribute weights.

6.3.2 Visualization and Semantic Interactions

The Visualization for this tool (bottom of Figure 8) simply consists of a large projection space accompanied by a column of attribute weights. Individual observations are still rendered as labeled nodes in the display, but are grouped by saturated convex hulls. The distance between pairs of observations is a weighted dissimilarity computation, in which the resting length of each link corresponds to the difference between the observation endpoints across all attributes.

Once again, analysts can perform OLI interactions on the observations. However, these semantic interactions only affect the mathematical models when an observation has been reclassified into a new cluster (i.e., manipulating the distance between observations within a cluster has no effect on the learned weights). When an analyst adds an observation to a cluster or removes an observation from a cluster (or both), the attribute weights are recalculated based on a dissimilarity measurement between the relocated observation and the centroid(s) of the involved cluster(s). After this computation, the resting length of each link is recalculated based on the new attribute weights. As a result, additional observations may reclassify themselves as the force-directed layout repositions the observations in the projection.

6.3.3 Pipeline

CSV Data Controller: The Data Controller used in this tool is identical to that used in Andromeda: numerical high-dimensional data is simply read in from a CSV file and normalized, and attributes weights are initialized to equal values.

Dissimilarity Model: The forward computation of the Dissimilarity Model computes a distance between each pair of observations, taking into account both the differences between the attributes values and the weights that have been learned for those attributes. This dissimilarity matrix is then passed to the Layout Model to be rendered. The inverse computation aims to understand why the analyst decided that this observation does not belong to its original assigned cluster and/or better belongs to its analyst-assigned cluster. This is accomplished by computing a distance between the observation and the involved cluster(s) centroid(s), ranking the attributes based on dissimilarity, and then updating the attribute weights accordingly.

Force-Directed Model: After a distance has been computed for every observation pair, these distances are loaded into a force-directed node-link visualization. The force-directed graph then stabilizes to a low-energy layout. There is no inverse computation for this model.

k-Means Model: Clusters are computed continuously using a modified *k*-means algorithm in the forward computation of the k-Means Model. This computation has been altered from traditional *k*-means to include a maximum cluster radius that allows some nodes to exist external to all clusters. As the force-directed graph reaches a stable layout, individual observations may transition into and out of clusters as they move closer to and further from each cluster centroid. The inverse computation of this model detects analyst-initiated changes in observation clustering assignments, and it passes the old and new cluster information to the next inverse computation.

7 Discussion

In this section, we discuss the implications of our new pipeline that is capable of capturing the complexity of semantic interactions in visual analytics tools. This discussion includes how this pipeline highlights the various semantic interaction possibilities in any given visual analytics tool, the ability to leverage this pipeline for rapid prototyping, and the limitations of this pipeline.

7.1 Exploring the Design Space of Semantic Interaction

With the greater emphasis placed on the mathematical models in our proposed new pipeline for visual analytics tools, opportunities for semantic interaction are highlighted. This is due to the fact that every Model should have both a forward computation and an inverse computation. If a given pipeline does not have an inverse computation for a Model, such as in the Sampling Model in the Piecewise Laplacian projection tool [22], then perhaps there is a missed opportunity for implementing a semantic interaction. Even for those that already have inverse computations, there may still be the potential to implement an additional or alternative inverse computation for the same Model.

For example, the forward computation in Andromeda's WMDS Model uses two parameters (the high-dimensional data and a set of attribute weights) to produce a single output (low-dimensional coordinates). However, the "inverse WMDS" computation described only computes new attribute weights given new low-dimensional coordinates, thereby assuming the high-dimensional data is static. However, what if instead the assumption was that the attribute weights were static and new high-dimensional data for an undefined observation was desired? Such an interaction may be triggered by the analyst clicking in an empty space of the projection not already occupied by an observation, effectively interpolating what attributes a high-dimensional observation would have if it were projected in that location.

Additionally, our previous research has identified a variety of ways to combine a similarity-based Model with a clustering Model to create different types of projections [30]. With this multitude of pipelines possible, there are naturally many methods of enabling semantic interactions with just two Models. Thus, in cases such as these, our newly-proposed pipeline can help further explore the design space of semantic interaction by highlighting the numerous possibilities, even when there are few Models involved.

7.2 Rapid Prototyping to Explore Design Trade-Offs

To quickly and efficiently explore the design space of semantic interaction, the ability to rapidly prototype several techniques from the visual analytics literature, and augment them with semantic interaction, would be immensely helpful. Trade-offs in different implementations may imply different Models being used or perhaps the same ones being altered to produce different results. For example, Cosmos is very similar in appearance to Andromeda, yet functions more like StarSPIRE (as evidenced by the similarities in their pipeline representations). SIRIUS and the visualization by Wenskovitch and North are both similar to Cosmos in their own manners as well, yet these tools operate in distinctly different manners by adding additional Models to the pipeline.

However, the current issue in experimenting with these kinds of trade-offs is that many visual analytics tools are created such that changing one Model for another is difficult; too often, the program structure for the given tool is heavily reliant on the specific Models being used. By defining the pipeline components and creating a pipeline such as the ones that we present here, we assert that our new visual analytics pipeline can help promote more modularized code. This is because every pipeline component has composability requirements, which are described by the arrows between the pipeline components. By structuring the program for the tool in this manner, the composability requirements help enforce modular code design. This modularity then makes interchanging different Models– and even different Visualizations and Data Controllers– trivial, thereby making exploring different areas of the design space of semantic interaction even easier.

For example, it may be apparent from Figure 6 and Figure 7 that Cosmos and SIRIUS have very similar-looking Visualizations. This is because the Cosmos pipeline– including its Models and Visualization– were all leveraged and adapted to enable SIRIUS. In fact, we have been able to separate each pipeline component to the point that we are able to interchange Cosmos and SIRIUS at will.

7.3 Limitations

Despite the power and flexibility of our proposed new visual analytics pipeline for semantic interaction, it is not without limitations. We briefly address several of these limitations here.

7.3.1 Requirements Limitations

The primary limitation of our semantic interaction pipeline lies in the requirement of providing an inverse computation for each forward computation. We assert this requirement as essential for enabling semantic interaction, yet we provide no guidance for how to determine what such an inverse computation should be. That is, the inverse computation can be mathematically rigorous, heuristic, or probabilistic, but the creation of the inverse computation lies with the Model creator.

Similarly, the pipeline requires Models to be composable with other pipeline components, but we provide limited instructions for defining this composability. For example, if the Text Data Controller for Cosmos were altered to use dynamic document sets, then at some point before any of the data preprocessing steps could be performed, the TF-IDF values would need to be computed on the fly. This would allow the other pipeline components to remain unchanged. If instead the TF-IDF values were not computed and no data preprocessing steps occurred in the Data Controller, the responsibility for doing so (to maintain data composability with the WMDS Model) would either fall to the Relevance Model or to a new Model that would rest between the Data Controller and Relevance Model.

7.3.2 Limitations of Pipeline Components

Another potential limitation is that we define a Model to be comprised of any forward computation and at least one accompanying inverse computation. It may very well be that there are only a few different categories or types of such Models (e.g., data manipulation Models that work the raw data into a form usable to the Visualization, projection Models that determine the overall type of projection used in the Visualization, and other Models that seek to augment the Visualization with additional information). Such a categorization may be useful to define the nuanced differences between how Models may be used and which ones definitely should have inverse computations to interpret semantic interactions. However, we do not attempt to make any such categorization; instead, we focus on the overall pipeline structure to generate discussion and critical thinking regarding which Models *should* be included, *how* the Models fit together to realize an interactive visual analytics tool, and the various manners in which semantic interaction *can* be realized.

7.3.3 Limitations of the New Visual Analytics Tools

Rather than creating fully-featured tools, we use this pipeline to quickly and efficiently prototype visual analytics tools to explore the semantic interaction design space. As a result of this design decision, the prototypes that we implemented in Section 6 only support a limited number of semantic interactions. However, we argue that each of our prototypes can support additional semantic interactions with the addition of more Models or alterations of existing Models in each pipeline.

8 Conclusion

In this work, influenced by the Sensemaking Loop described by Pirolli and Card [23] as well as the growing body of visual analytics tools that implement semantic interaction, we proposed three characteristics shared by semantic interaction applications: model composability, model inversion, and pipeline bidirectionality. From these characteristics, we proposed a new visual analytics pipeline that enables proper representation of the complexity involved in semantic interactions. This new pipeline is comprised of three main types of components: Data Controllers, Models (containing forward and inverse computations), and Visualizations.

We demonstrated the ability of our new pipeline to capture semantic interactions by providing pipeline representations of existing visual analytics tools. Then, we discussed pipeline representations for new visual analytics tools, highlighting the extensibility of this new pipeline new research in this area.

We also briefly discussed how this new pipeline may help further the exploration of the design space of semantic interaction and enable rapid prototyping of new visual analytics tools with semantic interaction. By rapidly prototyping such tools, researchers will be able to quickly create and study many alternative methods of semantic interaction. We intend to continue expanding on our own prototypes and conduct user studies to study how analysts perceive and use different visualization and interactions. Such research may uncover which methods best support the analyst's sensemaking process and how to develop better visual analytics tools in the future.

Acknowledgments

This research was partially supported by NSF grant IIS-1447416. The authors also wish to thank the BaVA @ VT research group for their contributions in this research, General Dynamics for their support, and the reviewers for helping improve this paper.

References

[1] S. Amershi, J. Fogarty, and D. Weld. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 21–30. ACM, New York, NY, USA, 2012. doi: 10.1145/2207676.2207680

[2] C. Andrews, A. Endert, and C. North. Space to think: Large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp.

55–64. ACM, New York, NY, USA, 2010. doi: 10.1145/1753326.1753336

[3] C. Andrews and C. North. Analyst's workspace: An embodied sensemaking environment for large, high-resolution displays. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 123–131, Oct 2012. doi: 10.1109/VAST.2012.6400559

[4] L. Bradel, C. North, L. House, and S. Leman. Multi-model semantic interaction for text analytics. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 163–172, Oct 2014. doi: 10.1109/VAST.2014.7042492

[5] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 83–92, Oct 2012. doi: 10.1109/VAST.2012.6400486

[6] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[7] E. P. dos Santos Amorim, E. V. Brazil, J. Daniels, P. Joia, L. G. Nonato, and M. C. Sousa. ilamp: Exploring high-dimensional spacing through backward multidimensional projection. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 53–62, Oct 2012. doi: 10.1109/VAST.2012.6400489

[8] M. Dowling, J. Wenskovitch, J. Fry, S. Leman, L. House, and C. North. SIRIUS: Dual, symmetric, interactive dimension reductions. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Oct 2018.

[9] S. M. Drucker, D. Fisher, and S. Basu. Helping users sort faster with adaptive machine learning recommendations. In P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, eds., *Human-Computer Interaction – INTERACT 2011*, pp. 187–203. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[10] A. Endert. Semantic interaction for visual analytics: Toward coupling cognition and computation. *Computer Graphics and Applications, IEEE*, 34(4):8–15, July 2014. doi: 10.1109/MCG.2014.73

[11] A. Endert, P. Fiaux, and C. North. Semantic interaction for sensemaking: Inferring analytical reasoning for model steering. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2879–2888, Dec 2012. doi: 10.1109/TVCG.2012.260

[12] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 473–482. ACM, New York, NY, USA, 2012. doi: 10.1145/2207676.2207741

[13] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 121–130, Oct 2011. doi: 10.1109/VAST.2011.6102449

[14] L. House, S. Leman, and C. Han. Bayesian visual analytics: BaVA. *Statistical Analysis and Data Mining*, 8(1):1–13, 2015. doi: 10.1002/sam.11253

[15] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2563–2571, Dec 2011. doi: 10.1109/TVCG.2011.220

[16] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. *Visual Analytics: Definition, Process, and Challenges*, pp. 154–175. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-70956-5_7

[17] H. Kim, J. Choo, H. Park, and A. Endert. Interaxis: Steering scatterplot axes via observation-level interaction. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):131–140, Jan 2016. doi: 10.1109/TVCG.2015.2467615

[18] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, Jan 2017. doi: 10.1109/TVCG.2016.2598446

[19] S. C. Leman, L. House, D. Maiti, A. Endert, and C. North. Visual to parametric interaction (v2pi). *PLoS ONE*, 8(3):1–12, 03 2013. doi: 10.1371/journal.pone.0050474

[20] G. M. H. Mamani, F. M. Fatore, L. G. Nonato, and F. V. Paulovich. User-driven feature space transformation. *Computer Graphics Forum*, 32(3pt3):291–299, 2013. doi: 10.1111/cgf.12116

[21] V. Molchanov and L. Linsen. Interactive Design of Multidimensional Data Projection Layout. In N. Elmqvist, M. Hlawitschka, and J. Kennedy, eds., *EuroVis - Short Papers*. The Eurographics Association, 2014. doi: 10.2312/eurovisshort.20141152

[22] F. Paulovich, D. Eler, J. Poco, C. Botha, R. Minghim, and L. Nonato. Piecewise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum*, 30(3):1091–1100, 2011. doi: 10.1111/j.1467-8659.2011.01958.x

[23] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. *Proceedings of International Conference on Intelligence Analysis*, 5, 2005.

[24] T. Ruotsalo, J. Peltonen, M. Eugster, D. Glowacka, K. Konyushkova, K. Athukorala, I. Kosunen, A. Reijonen, P. Myllymäki, G. Jacucci, and S. Kaski. Directing exploratory search with interactive intent modeling. In *Proceedings of the 22nd ACM international conference on Conference on information and knowledge management*, CIKM '13, pp. 1759–1764. ACM, New York, NY, USA, 2013. doi: 10.1145/2505515.2505644

[25] J. Z. Self, X. Hu, L. House, S. Leman, and C. North. Designing usable interactive visual analytics tools for dimension reduction. In *CHI 2016 Workshop on Human-Centered Machine Learning (HCML)*, p. 7, 05/2016 2016.

[26] J. Z. Self, R. K. Vinayagam, J. T. Fry, and C. North. 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*, HILDA '16, pp. 3:1–3:6. ACM, New York, NY, USA, 2016. doi: 10.1145/2939502.2939505

[27] F. M. Shipman and C. C. Marshall. Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems. *Computer Supported Cooperative Work (CSCW)*, 8(4):333–352, 1999. doi: 10.1023/A:1008716330212

[28] E. Wall, S. Das, R. Chawla, B. Kalidindi, E. T. Brown, and A. Endert. Podium: Ranking data using mixed-initiative visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):288–297, Jan 2018. doi: 10.1109/TVCG.2017.2745078

[29] X.-M. Wang, T.-Y. Zhang, Y.-X. Ma, J. Xia, and W. Chen. A survey of visual analytic pipelines. *Journal of Computer Science and Technology*, 31(4):787–804, 2016. doi: 10.1007/s11390-016-1663-1

[30] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, S. Leman, and C. North. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Transactions on Visualization and Computer Graphics Proceedings of the Visual Analytics Science and Technology 2017*, 24(01), January 2018.

[31] J. Wenskovitch and C. North. Observation-level interaction with clustering and dimension reduction algorithms. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, HILDA'17, pp. 14:1–14:6. ACM, New York, NY, USA, 2017. doi: 10.1145/3077257.3077259