Drag and Track: A Direct Manipulation Interface for Contextualizing Data Instances within a Continuous Parameter Space

Daniel Orban, *Student Member, IEEE*, Daniel F. Keefe, Ayan Biswas, James Ahrens, *Member, IEEE*, and David Rogers

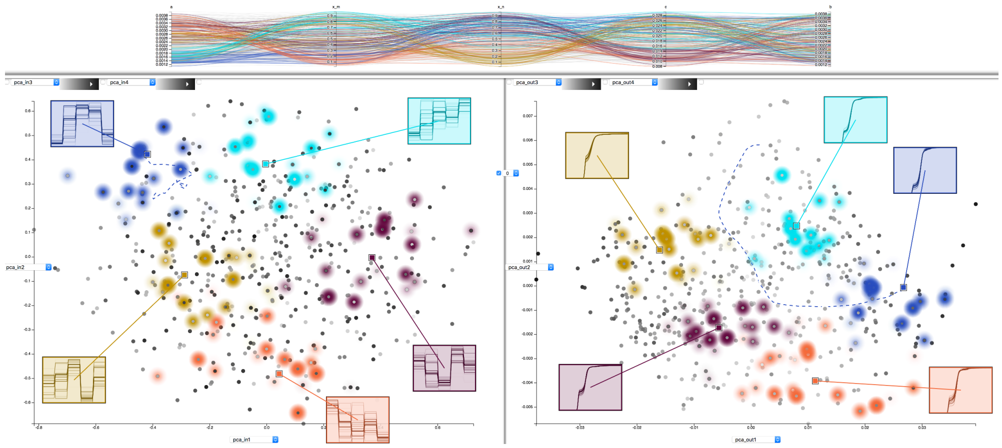

Fig. 1. Our parameter space exploration tool allows users to query linked dimensionally reduced spaces by interactive clicking and dragging virtual data instances. These are visualized via interactive callouts that can be directly manipulated to search and display the nearest ensemble instances. The linked parallel coordinate plot allows for filtering the ensemble and displays trends of selected parameters based on the user-defined annotation.

Abstract— We present a direct manipulation technique that allows material scientists to interactively highlight relevant parameterized simulation instances located in dimensionally reduced spaces, enabling a user-defined understanding of a continuous parameter space. Our goals are two-fold: first, to build a user-directed intuition of dimensionally reduced data, and second, to provide a mechanism for creatively exploring parameter relationships in parameterized simulation sets, called ensembles. We start by visualizing ensemble data instances in dimensionally reduced scatter plots. To understand these abstract views, we employ user-defined virtual data instances that, through direct manipulation, search an ensemble for similar instances. Users can create multiple of these direct manipulation queries to visually annotate the spaces with sets of highlighted ensemble data instances. User-defined goals are therefore translated into custom illustrations that are projected onto the dimensionally reduced spaces. Combined forward and inverse searches of the parameter space follow naturally allowing for continuous parameter space prediction and visual query comparison in the context of an ensemble. The potential for this visualization technique is confirmed via expert user feedback for a shock physics application and synthetic model analysis.

Index Terms—Visual Parameter Space Analysis, Ensemble Visualization, Semantic Interaction, Direct Manipulation, Shock Physics

1 Introduction

Simulation has become a vital part of the scientific exploration process, often replacing expensive and sometimes infeasible experiments. Models represent physical phenomena and can produce large sets of results, providing more data than ever before. Ensembles, concrete data sets whose instances are each characterized by a unique set of parameters [36], are widely used for visual analysis tasks including evaluating model uncertainty [24, 25, 33, 41], discovering trends [35], and making predictions [3, 40, 50].

Recent advances in interactive ensemble visualization tools have enabled new exciting modes for subject matter experts to explore their data. Mechanical engineers can directly manipulate 3D models to search a design space [14] and smoothly interpolate [44] between data instances. Flood management experts interactively make decisions by steering fluid flow simulations through multiple branching flood scenarios [54]. Graphics artists can inversely design animations by choosing their desired output from a set of clustered visuals [9, 32]. Scientists use dimensionally reduced (DR) linked views and interactive query widgets to annotate, cluster, extract features, explore, and modify their high-dimensional data [26, 39, 42, 52, 53]. Even in our daily lives, weighted views of ensembles can inform common, but difficult, multi-criteria decision making tasks like "Which car should I buy?" [37]

Although much work has been done, visualizing ensembles is still a long-standing problem, actively occupying recent developments in the visualization community. The call to action has only increased along with the demand to explore parameter spaces representing real-world scientific problems. Consider experiments in shock physics [7], where scientists endeavor to understand complex deformation properties of material under extreme thermodynamic conditions. Compressed gas is released from a gas gun to accelerate a projectile down a barrel, which collides with a target composed of a material of interest. The resulting shock wave traveling through the target produces time-series

• *Daniel Orban and Daniel F. Keefe are with the University of Minnesota, USA. E-mail: {dtorban, dfk}@umn.edu.*
• *Ayan Biswas, James Ahrens, and David Rogers are with Los Alamos National Labs. E-mail: {ayan, ahrens, dhr}@lanl.gov.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

rear-surface velocity curves, also called velocimetry profiles. These profiles provide valuable feature information regarding the elastic and plastic material properties.

Shock physics experimental techniques are about to become orders of magnitude faster, enabling the collection of extremely large amounts of parameterized data [6, 31, 56]. Currently scientists determine which tests to run based on experience and knowledge, but this will be inadequate as the next generation of experimental facilities come online. Instead of planning the next experiment, the scientist will be asked to plan the next hundred experiments. This is especially hard when scientists are unsure which questions are worth exploring in order to maximize the knowledge of a system while minimizing expensive experimental overhead. Fortunately, simulation models can help sample the space of possibilities and provide insight into the unknown by predicting velocimetry profiles. The hope is that material scientists can use a simulation ensemble to ask questions like, "What happens to this elastic feature if this subset of input parameters is explored?", "Which input parameters reduce the uncertainty in the plasticity?", or "What experiments should I run to reproduce this velocimetry curve?"

Our work is motivated by the modern approach for discovering information epitomized by the popular phrase, "Google it." Users simply type their question into a search engine as a text query and results ordered by relevancy are returned. The experience is simple, organic, and does not require an in-depth knowledge of the intrinsic structure required by most database engines. Another important feature that relevancy searches commonly employ is the fact that queries can be inaccurate. These systems often correct spelling errors or suggest similar queries (i.e. "Did you mean to search for ...?").

Imagine a visual search engine that allows users to find data instances in an ensemble by simply showing the search engine what they are looking for. In the shock physics application described above, a scientist may have a hypothesis that a certain velocimetry profile is created when parameter x is high and parameter y is low. To find out whether this is a reasonable assumption, the scientist can use the visual search engine to retrieve similar data instances by suggesting an approximate velocimitry curve and a high x parameter. The results can be analyzed to see if, in fact, y is statistically low. In order to change the query, the user simply changes the curve. One way this is done for interactive visualizations is to use direct manipulation techniques, which allow users to directly modify data through representative visuals [14, 16, 34]. These organic and qualitative scientific inquiries motivate interactive solutions to the following three general ensemble visualization problems:

Problem 1: Can forward and inverse prediction approaches be combined into one approach that uses both input and output parameters? Ensembles are calculated from an input-output model, a function that maps input parameters to output parameters [45], which is typically a simulation model. In relation to inputs and outputs, research focuses on either 1) forward approaches [44, 54], which use the input space to predict results and determine sensitivity, or 2) inverse approaches [9, 32], which use output parameters to fit input parameters. Similar to just a few systems that have been presented thus far in the literature [3, 13, 14], we ask whether forward and inverse approaches can be combined to creatively search parameter spaces.

Problem 2: High-dimensional projections are hard to interpret. Simulation data sets usually contain many input and output parameters. Due to the physical limitations of displays and our own visual system, it can be challenging to effectively visualize data that has greater than two or three dimensions [30]. One way to visualize these complex multi-dimensional spaces is to use dimensionality reduction techniques, which project high-dimensional data into fewer dimensions that more adequately represent the intrinsic structure of the data. Unfortunately, there is a known "trade-off between the interpretability of the axis and intrinsic structure captured by dimensionality reduction methods." [30] In other words, the visual representation of these dimensionally reduced spaces are usually hard to understand.

Problem 3: Estimation methods produce uncertain results. Scientists would like to interactively view and visually understand the gaps in continuous spaces without the expensive computation time required by simulation. Fortunately, estimation methods are able to predict results using the statistical properties of ensembles. However, ensemble parameter spaces are often sparse and statistical phenomena like the sampling density can cause uncertainty [3]. Therefore, estimates need to be displayed in the context of the ensemble for validation. This is especially true for inverse problems, which try to determine the input from the output. For these problems prediction actually represents a probability distribution and there is no "best" estimate, but rather multiple possible estimates [49]. How should we evaluate and adequately use estimates as they relate to navigating sparse parameter spaces?

Although simple and intuitive, there is one fundamental flaw with the search engine approach described above: users need to know what they are looking for. This assumption is not always true if the user is in exploration mode. What does it mean to search for "something interesting"? Often, models like those in the shock physics application, are non-linear and simple parameter relationships can be difficult to discover. Visual analytics approaches like dimensionality reduction interaction techniques help visualize the ensemble as a whole and understand trends in the data [42]. In the literature this navigation strategy difference is described as *local-to-global* vs. *global-to-local* [45]. We can either start from a data instance to find others (local-to-global) or start with an overview and filter down to relevant data instances (global-to-local). We work towards addressing visualization problems 1 and 2 above by tightly integrating these two navigational strategies.

As problem 3 states, estimates may not be completely accurate, however, their utility in navigating a continuous space is still valid when searching for similar instances. Queries for similarity searches can be approximate. In our approach, we use estimates as initial guesses and allow the user to refine them through direct manipulation. The similarity search provides the actual valid data instances that the user is looking for, which are accurate. Naturally, the returned data instances represent an uncertainty distribution that can be used to evaluate the prediction.

In this paper we present Drag and Track, a direct manipulation interface for contextualizing ensemble data instances within a continuous parameter space. We combine direct manipulation search techniques (drag) with ensemble dimensionality reduction methods to annotate (track) parameter changes as an ensemble is being explored. Our approach uses multiple virtual data instances (VDI) to ask "what if" questions. These user-defined virtual data instances have the same parameters as regular data instances, but also can be edited. Ensemble data instances along with VDIs are spatially positioned in dimensionally reduced (DR) linked views. VDIs are visualized as interactive callouts, which visually represent the data instance parameters. When the parameters in these callouts are directly manipulated, the underlying VDI is updated and acts as input to a similarity search that returns the nearest ensemble neighbors. The relevant data instances are highlighted, allowing users to annotate the low dimensional space and therefore contextualize data instances based on user-defined goals. As interactive callouts are manipulated, the VDI's movement is visualized as separate paths on the dimensionally reduced spaces, therefore tracking parameter sensitivity across the spaces.

We apply our approach to a real-world shock physics application, and evaluate it based on expert material scientist feedback. This paper makes the following contributions.

- A visualization technique that combines forward and inverse direct manipulation approaches with multiple linked dimensionally reduced views of an ensemble to visually understand an annotate a continuous parameter space.
- A visual similarity search approach that uses the direct manipulation of user-defined virtual data instances to find nearest neighbor distributions of an ensemble.
- A visual approach for qualitatively tracking changes in parameters across multiple linked dimensionally reduced views using continuous estimates.

- A mechanism for specifying user-defined annotations of dimensionally reduced spaces using the results from multiple similarity searches.
- An evaluation of the usefulness of the visualization based on expert material scientist feedback and synthetic model analysis.

2 Related Work

Our research is derived from visual parameter space exploration and other related ensemble visualization techniques. We also extend work done in and direct manipulation and visual analytics interfaces.

2.1 Visual Parameter Space Exploration

Parameter space exploration involves analyzing, optimizing, and estimating parameter relationships in an ensemble data set. Sedlmair et al [45] provide an in depth literature review of visual parameter space analysis tools with a conceptual framework that can be used to evaluate new and existing approaches. The framework describes the navigational strategies *local-to-global* and *global-to-local*. Local-to-global allows users to search a space one instance at a time [1, 14], while global-to-local starts with an overview of the instances and allows users to drill down to instances of interest [9]. In our approach we use both strategies. Multiple virtual data instances (local-to-global) are viewed in the context of the ensemble (global-to-local) and represent a weighted distribution of ensemble instances.

In this area, our work most closely resembles Design by Dragging [14], a direct manipulation interface that allows users to interactively change both input and output parameters in order to navigate a set of CAD designs. Although we employ a similar mechanism for exploration, Design by Dragging only allows for navigating between two similar instances, and only shows one estimate at a time. We visualize multiple estimates and allow for prediction models that can interpolate between several instances.

As part of Sedlmair et al's framework, fast *surrogate models* are often used to predict results based on interpolation, regression, or machine learning. This type of estimation enables uncertainty analysis, optimization, and parameter fitting [45]. Our work is similar to Berger et al's [3], which uses regression and nearest neighbor prediction approaches to use a sampled parameter space as a continuous multidimensional function. Our approach, however, also allows for multiple estimations in the context of the ensemble for visual comparison.

Special care needs to be taken when using estimation methods since surrogate models may be less accurate for sparse parameter spaces [3]. Tuner [50] optimizes image segmentation algorithms by comparing Gaussian process estimates with ground-truth images. A Gaussian process provides uncertainty information with estimates, enabling the ability to determine areas of interest. For example, it may be advantageous to add more samples in an area which has a high uncertainty and desired estimates to optimize the expected gain [50]. HyperMoVal [40] also uses estimation with support vector regression. It compares the surrogate models themselves to find regions of best fit and validate their physical plausibility. Our estimates are qualified in the context of local distributions providing parameter uncertainty information based on similarity.

We also want to explore a space based on both input and output parameters. One of the most common approaches is to project the output space into 2D images [1, 9, 32, 38]. These images can be used for feature extraction [2], clustering [9], optimization [14, 32], evaluation [50], and navigation [1, 14, 23]. Other output representations focus on computing correspondences between instances in an ensemble. Schulz et al [44] store pre-computed samples of CAD models with various combinatorics into adaptive grids, enabling instance interpolation. Yumer et al [55] present an approach that ties computed deformation handles across similar shapes to crowd sourced parameters, making it possible to modify geometry based on interpolated semantic properties. Dutta et al [17] use Gaussian mixture models to generate feature correspondences across time varying simulations, allowing a distribution based analysis approach. We provide 2D views of the input and output parameters, while maintaining a tightly coupled relationship to the underlying data for inverse queries.

2.2 Ensemble Visualization

Parameter space exploration is a type of ensemble visualization where we search for and analyze parameter relationships. This broader class focuses on understanding variation across multiple related datasets, which are usually characterized as being large, temporal, multidimensional and multivariate. The applications for ensemble visualization also vary significantly. Some approaches work towards mitigating uncertainty and error in simulation results that arise from misunderstanding real-world phenomena [3, 5, 29, 40, 41]. Others use dimensionality reduction (DR) techniques or interactive widgets (i.e. graphs or parallel coordinate plots) to view very high-dimensional spaces, allowing feature based views of their data [4, 26, 30, 53]. Cluster analysis is also used to partition subsets of ensembles with localized statistical properties, providing the ability to compare trends and understand local variances [9, 24, 25, 28].

Specifically in our work, we use DR projections to visualize ensembles in feature spaces, and use visual querying to represent localized weighted sample distributions. This enables us to evaluate our predictive models in the context of the ensemble. For a detailed survey on dimensionality reduction methods as it applies to high-dimensional data, we refer the reader to Van der Maaten et al [51] and Liu et al [30]. In our examples we use principal component analysis (PCA), which finds projections whose components are ordered to maximize the variation, and multidimensional scaling (MDS), which uses the distances of individual data instances to show the level of similarity.

Our visual exploration tool resembles Steiger et al [48], who visualize clusters of time-series data on a DR view and display the mean at each cluster via a visual callout. This tool also shows paths on the DR space to show how a specific instance has changed over time. Similarly we use visual callouts, but also allow users to define and move them around interactively to see relationships between DR spaces. Additionally, our callouts show estimates that are qualified by uncertainty distributions and can be directly manipulated, changing the way a DR space is explored.

2.3 Direct Manipulation for Visual Analytics

This paper is inspired by direct manipulation interfaces, which allow users to interact with geometry [14] or visual data representations [13, 23] to navigate or affect a system. Accessible Animation [34] enables users to explore and modify animation keyframes by "tugging" directly on the animation. Similarly, through Video Browsing [16] it is possible to inversely change frames on a video by clicking and moving objects around on the screen. Design by Dragging [14] moves these techniques into the scientific domain. Users can either change geometrical dimensions (forward design) or drag stress calculated from a finite element method (FEM) around on a mesh (inverse design). These approaches fit well into our natural experience and also enable exciting ways to understand parameters that are related to interesting phenomena. We apply similar forward and inverse direct manipulation approaches, but extend them to virtual data instance interactions.

The visual analytics community continues to invest in direct manipulation and inverse techniques. Sacha et al [42] provide a detailed survey of interaction techniques applied to DR. The authors list a seven guiding scenarios for DR interaction. Of these, data manipulation, which allows users to directly manipulate data points similar to interaction in our system, was rarely used. A recent paper by Cavallo and Demiralp [13], however, shows the utility of such interaction, especially as how it relates to understanding DR. They provide an impressive framework that explores interacting with forward and backward projection mappings for both PCA and audoencoders. Users can move data points around to see how parameters change and also directly manipulate output images to see how the point moves in the DR space. Our forward and inverse interaction techniques are similar when exploring a DR space, as both approaches use out-of-sample extensions [51], which allow new data points to be projected into an existing DR. However, we are also interested in understanding the relationship between multiple spaces, an important feature when understanding model sensitivity. We add the extension of being able to manipulate multiple data points that are not part of the original ensemble, allowing the user to explore multiple parameter relationships at the same time. This enables user-defined comparisons between areas of interest across DR views, and the development of custom illustrations that represent the parameter space as a whole.

Our approach falls into the category of semantic interaction [18, 19, 21], which focuses on using human cognition to steer the underlying computations through the direct manipulation of spatializations. Many visual analytic tools take advantage of spatial relationships or relative proximity to represent similarity. For example, Dust and Magnets [47] adds the metaphor of using anchor points as magnets that attract data points. Similarly, ForeSPRIE [20, 21] allows users to drag documents around a force directed screen to enable user intuition based clustering. Visualization by Demonstration [43] suggests specific visualizations based on a user's interaction like positioning data points or changing their size or color. Contrasted with syntactic interaction, which requires that the user know about the complex algorithms [18], semantic interaction tries to change the distance function weighting [8] or DR method [22] to optimize meaning. Since we take advantage of data point approximations to intuitively understand a parameter space, Drag and Track can be considered a semantic interaction technique. Most semantic interaction tools change the DR mechanisms to improve understanding. We feel that we can naturally incorporate these techniques into Drag and Track as described in Section 9, however, we focus specifically on intuitively understanding the DR space and how DR spaces are related to each other without modifying the spaces. We aim to understand the parameter space from the direct manipulation of high-dimensional data in a natural and intuitive way.

3 Overview, Example Problem, and Definitions

The Drag and Track framework allows users to creatively annotate a set of dimensionally reduced spaces through user-defined virtual data instances. Our approach requires four pre-conditions:

1. An input-output model, typically a simulation model, that defines a set of parameterized simulation runs we call data instances. The ensemble being studied is a set of data instances.
2. A set of dimensionality reduction (DR) methods that describe the spaces to explore. For example, in our analysis we use principal component analysis (PCA) and multidimensional scaling (MDS) as DR methods.
3. A distance metric between data instances and a similarity search method, both used for finding nearest neighbors in an ensemble.
4. An estimation method, which takes a subset of input and output parameters and returns a predicted data instance.

Example Input-Output Model. Figure 3 shows an example parameterization of a shock physics simulation using the Johnson-Cook strength model [27] given as

$$Y = \left[A + B\varepsilon_p^n\right]\left[1 + C\ln\left(\frac{\dot{\varepsilon}_p}{\dot{\varepsilon}_0}\right)\right]\left[1 - \left(\frac{T - T_{\mathrm{ref}}}{T_{\mathrm{melt}} - T_{\mathrm{ref}}}\right)^m\right], \quad (1)$$

The figure shows visual representations of a data instance for both input and output parameters. The left shows the input, where A,B,C,n, and m are the input parameters in Equation 1. The right shows the the result of the simulation as an output curve, along with the features the scientists are interested in. Users are interested in how the input parameters affect the features in the output curve and the inverse. For example, material scientists might want to know what happens to the elastic plateau as A is increased. We now describe the Drag and Track technique using the example in Figure 2.

Dimensionally Reduced (DR) Spaces. In the figure, we start with two dimensionally reduced linked views of the ensemble. The two views can be any dimensionally reduced spaces, however in this paper we focus on the input and output spaces as example spaces that help describe our approach. These two spaces are conceptually concrete and are directly applicable to analyzing sensitivity in the shock physics application. Although our approach can scale to several DR spaces, we assume two for our examples.

Virtual Data Instances (VDI). The figure shows each ensemble data instance as a point in a scatter plot. Virtual data instances are shown as squares, and they represent user-defined data instances positioned in the DR space using each view's DR method. VDIs have the same parameters as regular ensemble instances, but their parameters are editable. Figure 2 shows linked interactive callouts that visually display a subset of VDI parameters that are relevant to the DR view. In this case, the callout in the input space shows the input curve, and the output callout shows the output curve.

Linked Interactive Callouts. Linked interactive callouts can be directly manipulated to change the parameters in the VDI they represent. For example, in Figure 2 the user drags the center of the input curve up, changing the VDI and the its location in both DR spaces. Finally, Figure 4 shows how multiple VDI's annotate the parameter space by highlighting similar instances in the DR spaces.

System Overview. The Drag and Track work flow tightly integrates user interaction techniques like direct manipulation with similarity search and data instance estimation techniques. Figure 4 shows the technical framework describing the visualization interface. Virtual data instances act as the glue between user interaction and back-end processes. Users can annotate DR spaces by creating VDIs, which in turn can be directly manipulated. VDIs hold query information related to parameters of interest, and when interactive callouts are moved or changed, the query is updated and a similarity search is invoked, which provides a set of nearest neighbors. Finally, as in Figure 4, an estimation step updates the VDI's predicted data instance, which affects the visual callout and the positions of the VDI in the DR spaces.

In the following two sections, we describe our technical framework, first from the user interaction perspective (Drag), and then from the visual analytics perspective (Track). Drag includes tasks like direct manipulation and positioning VDIs inside the DR views. Track involves calculating and displaying similarity search results with predictions to ultimately annotate the parameter space. After describing the general framework in Sections 4 and 5, we provide a concrete implementation in Section 6.

4 Drag: Direct Manipulation of Virtual Data Instances

The primary method for user interaction in the Drag and Track framework is to update virtual data instances. There are two modification modes:

1. Annotate parameter spaces by creating or dragging the VDI points on the dimensionally reduced space.
2. Directly manipulate callouts to change parameters in the VDI.

Although a VDI looks like a regular data instance, the data structure itself is a bit more complicated as detailed in Figure 4. A VDI contains a query set, a predicted data instance, and a set of nearest ensemble neighbors. The query set is a subset of input and output parameters that contain parameter values and weights. This data structure is used for the similarity search step and estimation step shown in the figure. The nearest neighbors are the output from the similarity search, and the predicted data instance is calculated from the estimation step. The predicted data instance has the same parameters as regular ensemble data instances. Interactive callouts display the components of the VDI, and the VDI positions on the DR spaces calculated from the predicted data instance and the user's interaction.

Every data ensemble instance and predicted data instance also contains a set of derived parameters that are calculated from projecting the data instance to the dimensionally reduced space. For example, a data instance projected to both the PCA input and output spaces would have four derived parameters, each representing one axis from the two 2D spaces. These parameters are available in the VDI's predicted data instance and can be used as weighted query parameters.

4.1 Annotate Parameter Spaces

Users can interactively annotate the parameter space by creating multiple VDIs. This is done by simply clicking on any DR space at the locations the user would like to annotate. As shown in Figure 2, the VDI

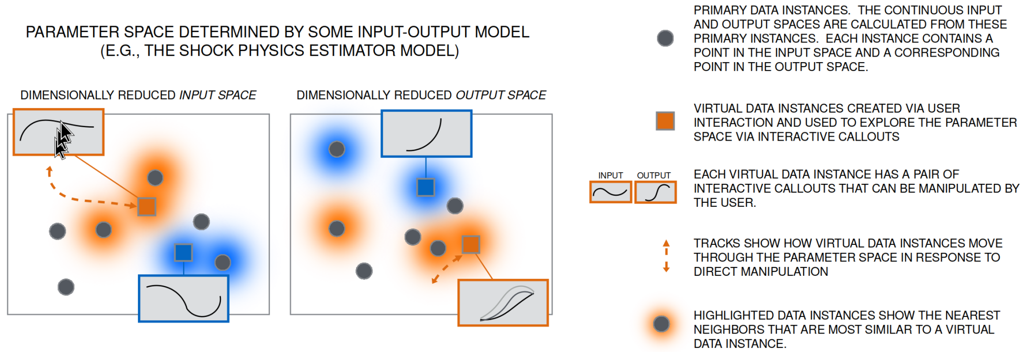

Fig. 2. Diagram explaining how direct manipulation is tracked on dimensionally reduced views. Virtual data instances are displayed as linked interactive callouts that are positioned in both the dimensionally reduced input and output spaces. After dragging to manipulate a parameter via the interactive callout, the corresponding VDI points move in both spaces and a track line shows how they travel through the space.

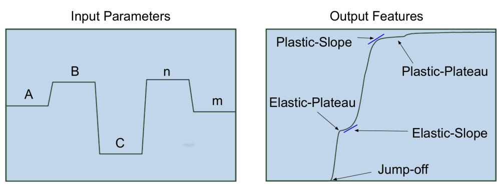

Fig. 3. A visual representation of the input and output of a shock physics simulation. The input parameters (left) are the A,B,C,n, and m parameters in Equation 1, and the output parameters (right) is the velocimitry profile.

shows up on all DR spaces at the calculated VDI position (described below). Users can then move the VDI points around any DR space to see how movement in one space changes the position in another space. Figure 2 illustrates the tracked path that is displayed in both spaces when a VDI changes. These paths are calculated from iterative position estimates as users move points from their start to their goal. Annotation comes from the highlights provided by the calculated nearest neighbors that are displayed across all DR views. Users can customize the VDI through direct manipulation to creatively highlight points in the space that match their specific investigation.

When a point is created or moved, the two dimensions on the selected view are added to the VDI's query set. As described in Section 5, the similarity search and estimate steps update the VDI's nearest neighbors and predicted data instance respectively. The nearest neighbor highlights, VDI position, and visual callouts are then updated. Reprojection of the VDI to the lower dimensional views is handled by the estimation step described in Section 5.2. The predicted data instance includes these estimated projected axes values. Here the user's lower dimensional query parameter values (i.e. the location the user clicks) and the predicted values may not match, causing an ambiguity on where to position the VDI on the dimensionally reduced spaces. Our approach is to present both sets of values and allow the specific implementations of the Drag and Track framework to interpolate between them. This design decision is discussed in detail in Section 9.

Our framework also integrates a linked parallel coordinate plot as displayed in Figure 1. The parallel coordinate plot lines, each representing a data instance, will highlight with the color of the user-defined VDI if it is a nearest neighbor. This enables users to see parameter associations between VDI regions and how they relate to the lower dimensional spaces. The parallel coordinate plot also allows filtering of the data set. The VDI points then describe new highlighted nearest neighbors and the estimate will be based on the filtered set. As described in section 7.2.2, this type of filtering is important in helping scientists constrain the ensemble to investigate sub-areas in the parameter space.

4.2 Directly Manipulate Callouts

On each DR space, every VDI point also has a space specific interactive callout as shown in Figure 2. The callouts are linked together across spaces through the VDI. These are different view related pictures of the ensemble parameters. Interactive callouts can use any information from the VDI to visualize the data including the query set, the predicted data instance, and the nearest neighbors. For example, referring to Figure 3 for the shock physics application, we display estimated inputs as step lines on the input callout and show the velocimetry profile curve estimate for the output callout. Since these are 1D curves, we also superimpose the nearest neighbors giving insight into the uncertainty in the distribution.

Interactive callouts are powerful because they tightly couple direct manipulation with the visualization. Like magic, users can drag a visual representation of a parameter and see the callout and the VDI position immediately change. For example, material scientists can click on the elastic plateau on the velocimetry profile and pull it up and watch other features like the plastic plateau change. In doing this, the user is asking the question, "What if the elastic plateau is higher?"

When a callout is directly manipulated, the parameter values related to the interaction are added to the query set along with parameter weights. Application specific callouts define how the visualization and interaction are related to the query set. In Section 6 we describe an example visual callout type and weighting scheme. As described in Section 4.1, a new similarity search is performed based on the updated query set and a predicted data instance is calculated. These feedback into the visualization loop by changing the VDI's positions and refreshing its visual representation in the interactive callouts.

5 Track: Using Virtual Data Instances to Visually Analyze Parameter Spaces

The Track part of Drag and Track involves the back-end algorithms necessary to enable the visual analytics of the parameter space. Figure 4 shows the Track actions on the right side of the diagram and they include the Similarity Search Step and the Estimation Step. Both are necessary to enable the visualization piece described in Section 4.

5.1 Similarity Search Step

The nearest neighbors search is a vital part of our framework since the nearest neighbors provide the highlights for the annotation in the visualization. Differences in customized VDIs can be tracked across the DR spaces while providing uncertainty information about the specific query set and predicted data instance. Figure 4 shows the search step being invoked when the query set in a VDI is updated. After the nearest neighbors are updated, the estimation step is called.

The similarity search requires a query point, $q \in \mathbb{R}^p$, a set of data points, $S \subseteq \mathbb{R}^p$, and a distance metric, $d : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, to calculate similarity where p is the number of parameters being compared. The query set defined in the VDI has a list of parameters, their values, and

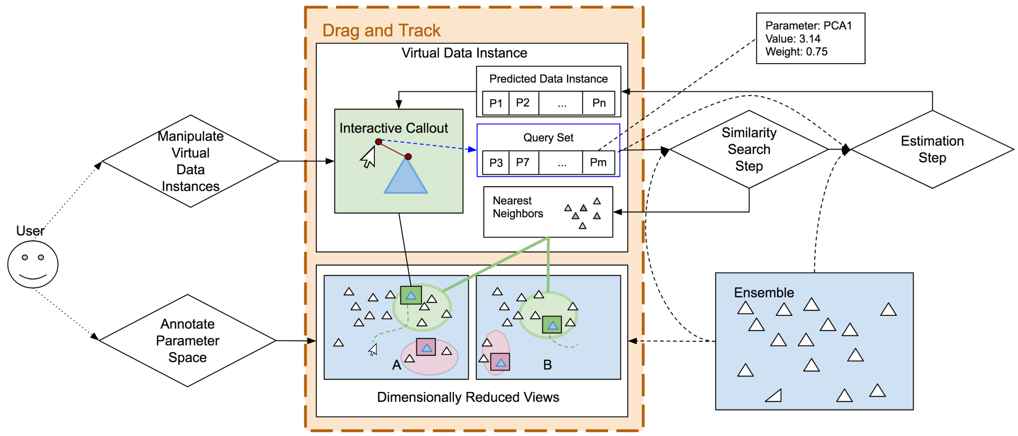

Fig. 4. Overview of the Drag and Track framework as it relates to user interaction influence over the similarity search and data instance estimation processes.

their weights as shown in Figure 4. The size of the query set determines p, and the query set elements determine the parameters for the data points created from data instances. The query point, q, is created by taking each parameter in the VDI query set and listing its value. Similarly, a point for each data instance, $x_i \in Ensemble$, is created by listing the corresponding parameter values to create a point $s_i \in S$. A weight vector, $w \in \mathbb{R}^p$, is also created by listing the corresponding VDI's query set parameter weights defined by the application.

The distance function $d(A,B)$ can be any measure of similarity between these points. The nearest neighbors are calculated by using the distance function to determine the difference between every $s_i \in S$ and the query point, q, as $y_j = d(s_j, q), j \in \{1...n\}$, where n is the size of the ensemble. The points in S are sorted by the vector y and the top k are used as the new VDI nearest neighbors.

5.2 Estimation Step

Finally, the last step in the track process is to provide a predicted data instance based on the query set in the VDI. This is important because we use the predicted data instance to determine the positions of the VDI in the DR spaces. The prediction is visualized in the VDI callouts to provide a starting point for the direct manipulation described in Section 4.2. The tracked lines on the dimensionally reduced spaces are also calculated from estimate iterations as parameters are manipulated.

Estimation methods include regression, interpolation, statistical emulators, or machine learning methods. For example, a subset of ensemble visualization tools incorporate fast *surrogate models* [45], which combine the statistical properties of ensembles and estimation techniques to predict results from an input-output model [3, 14, 40, 44, 50]. In this case, a surrogate model is defined as $\hat{f} : A \rightarrow B$, where A is the input parameter set and B is the output parameter set, such that $\hat{f}$ approximates a more complicated function f representing the input-output model. We relax the requirement that an estimator must go from input to output by defining it as any function that takes a VDI query set and returns a predicted data instance, which includes both and output parameters.

Finally, the estimation step re-projects the predicted data instance onto the dimensionally reduced spaces using out-of-sample extensions [13, 51] of the existing DR methods. If out-of-sample extensions are not available for the DR methods, we use the estimated values in the predicted data instance. The end result is a predicted data instance that is added to the VDI, which can then be visualized and qualified by a nearest neighbor distribution.

6 Implementation

We implemented the Drag and Track framework to study the parameters of 1D output curves as in the shock physics input-output model. Two linked scatter plots display the dimensionally reduced views of the input and output spaces as shown in Figure 1. As described in Section 3, implementations need to first provide four components: an input-output model, a set of dimensionality reduction methods, a distance metric with a search method, and an estimation method.

Our implementation allows for any input-output model where the input is a discrete set of quantitative parameters and the output is a 1D curve. Interactive callouts are displayed as 1D curves representing the predicted input or output. Each point on the callout's horizontal axis represents a separate parameter. To facilitate direct manipulation, users can click and drag points on a curve up and down. When the user clicks on a point, the horizontal axis parameter is added to the VDI's query set and updated with the curve's value at that point. Dragging the point up or down also updates the value for the parameter in the query set. Since the output is a 1D curve, we superimpose the curves of the nearest neighbors to show an uncertainty distribution along with the more prominent prediction curve.

The implementation also provides both PCA and MDS dimensionality reduction methods for the estimation step to use. When a user clicks on a scatter plot, a VDI is created and the x and y values are added to the VDI's query set for the specific derived dimensionally reduced parameters described in Section 4.1. The similarity search step and estimation steps are invoked and the result is displayed as VDI points and visual callouts in both dimensionally reduced spaces. When a user drags a VDI point, the same DR parameters are updated in the query set, and the algorithm is repeated.

In our implementation, multiple parameters can be added to a VDI query set and their weights are systematically updated after each interaction. We consider the most recent interaction to have the most weight. After each manipulation, the weights of previously added parameters are reduced, allowing the VDI to be naturally constrained based on several user-interactions without over-constraining the parameters. Dragging a VDI point in a DR space is considered a more drastic change than manipulating one variable. In this case, we increase the dimensional axis parameter weights and substantially lower the existing weights on the other query parameters.

For our similarity search, we look for the top-25 neighbors by comparing each data instance point with a query point calculated from the VDI query set. The distance metric we used is the weighted Euclidean distance. For the estimation step, we use the multidimensional form of inverse weighted distance interpolation [46] using 5 neighbors. Since the similarity search step is already performed, we can use the inverse distances of the nearest neighbors that were calculated using the VDI query set. The result is a predicted data instance that uses a percent of each nearest ensemble data instance based on how close the query point is to the data point.

Finally, VDI positions are calculated based on the interaction type. This is to address the ambiguity of a VDI's position mentioned in Section 4.1. To avoid confusion, when a VDI position is directly manipulated, the VDI's position is set to the query set value, which represents the user's desired location. For all other interactions we use the VDI's predicted data instance value as its position. Our solution to this problem is discussed in Section 9.

To analyze the Drag and Track framework, we use our implementation to investigate two separate application domains. First we apply it to a real-world high-dimensional shock physics problem (Section 7). Here we provide a set of interactive tasks by giving an example work-flow. We then evaluate usefulness Drag and Track via known information and expert user feedback. For the second application, we analyze the effectiveness of using Drag and Track across several synthetic models (Section 8).

7 Application to Shock Physics

Our Drag and Track solution was applied to a set of simulation reproductions of a shock physics experiment involving aluminum alloy Al-5083, specifically Shot 104S from Botler and Dandekar [7]. Aluminum alloys are used in many applications including baseball bats, transportation, tent poles, armored vehicles, and aircraft. Different elemental compositions of aluminum with other metals provide a variation of useful material properties. Due to its high strength and resistance to corrosion, Al-5083 is a preferred material used in many military and commercial applications including boat hulls, missile containers, military vehicles, and storage tanks for jet fuel [7].

In some thermodynamic conditions due to pressure, temperature or volume, Al-5083 may deform irreversibly, causing a plastic deformation. This makes the aluminum alloy not suitable for some applications, and therefore it is important to understand and predict the alloy's limitations. Shock physics experiments test these extreme conditions by using the pressure in gas guns to accelerate a projectile at the material of study. These experimental configurations are called shots, and their results can be analyzed to investigate the material. Simulations are used to try to approximate the experiment and understand which inputs might cause these plastic deformations.

Figure 3 shows the output for these experiments, called velocimetry profiles. The various features on the curve like the jump-off, elastic plateau, or plastic slope describe desirable and undesirable deformation properties. The input for the simulations represent variables in the Johnson-Cook strength model given as Equation 1. These variables can change the model to represent different thermodynamic conditions. The goal is to qualitatively understand the material properties of Al-5083 and the deformation causes based on the simulation model.

7.1 Example

This section describes an example usage scenario of our system and the tasks that would be used to evaluate its effectiveness. Sonja is a material scientist studying the material properties of aluminum with a new set of 500 simulations based on the Johnson-Cook model. She opens up our implementation of Drag and Track and loads her ensemble hoping to quickly understand some of the major parameter trends worth investigating (Work-flow shown in Figure 5). The first thing she does is try to explore the parameter space as a whole by clicking on seven different locations in the output space on the right (see Figure 5-(a)). Sonja observes that there are two major features that describe the output space, the elastic-plateau and the plastic-plateau. She also sees that, although there is a trend in the input space, the nearest neighbor highlights representing each VDI are very spread out, meaning that for each output curve there are multiple possible inputs. However, on the interactive parallel coordinates plot she notices that based on her initial annotation the A and m variable are both low for the output curve that is furthest right in the output space, implying there is an input parameter relationship for that curve.

Sonja finds one of the curves that look interesting and wonders how that curve's features are distributed throughout the spaces. She clears the screen and clicks the output space locating the curve of interest (see Figure 5-(b)-1). In order to understand how the elastic-plateau changes in both spaces, she clicks on the elastic-plateau on the visual callout and drags it up (see Figure 5-(b)-2). A path is interactively drawn across both the output and input spaces, informing her that the output DR space's left and right align with the height of the elastic-plateau. Similarly, she can observe that the input space is also correlated, but on a diagonal. Finally, she directly grabs the plastic-plateau and observes that a spatial relationship also exists in both spaces (see Figure 5-(b)-3). The highlighted data points are displayed in the input callout and the parallel coordinate plots, which show that the manipulated velocimitry curve is associated with a high C value and a low n value. This interaction can help her in the future when she is trying to fit an output curve generated from an experiment to the simulation output.

Sonja then wants to understand more deeply the sensitivity of her model for her original curve of interest. She clicks on the curve's position in the output space five times and notices that the data instance highlights and the VDI positions are all the same in both spaces (see Figure 5-(c)-1). In order to sample the input space more completely, she moves the VDI points to several different locations in the input space (see Figure 5-(c)-2). This will give her an idea of the various categories of inputs that cause the specific output curve. From the visual callouts in the input space and the parallel coordinate plot, she notices two relationships immediately. First, all of the inputs show a fairly low A value, meaning that a low A is associated with the specific curve. Second, the bottom right callout in the input space, which has the highest set of A values, requires a low C value. Sonja decides that she would like to continue her investigation by using the parallel coordinate plot to filter out high A values and low C values (see Figure 5-(c)-3). Fortunately, her preferences were saved in the various VDI's, so the positions and highlighted instances are updated to fit her user-defined illustration. We now describe our results using data from a shock physics simulation.

7.2 Data and Results

A shock physics simulation ensemble was generated using the hydrodynamics code FLAG developed at Los Alamos National Laboratory [10–12]. The data was computed using the Los Alamos National Lab's Moonlight supercomputing platform, which consists of 308 Intel Xeon E5-267 nodes: 4928 CPU cores. Besides the parameters listed in Figure 3 for the Johnson-Cook model, the FLAG simulation model also need a shear modulus and an impact velocity. We used the properties of Shot 104S [7] to provide these two values. Out of 1000 simulation runs generated, our analysis is based on a randomly sampled ensemble of 500. The ensemble represents 500 data instances with the Johnson-Cook model as input parameters and the velocimetry profiles as output parameters. In Section 7.2.1 we evaluate our approach based on known information about the ensemble, and in Section 7.2.2 we describe expert user feedback.

7.2.1 Parameter Sensitivity Analysis

The domain experts a performed sensitivity analysis on the full data set described above. The results of the sensitivity study show a high response for the relationship between the A input parameter and the Elastic Plateau, both shown in Figure 3. We wanted to confirm whether this relationship shows up using Drag and Track.

In order evaluate, we first varied A in the input space to see if there was a correlation to Elastic Plateau shown on the interactive callouts on the output space. Second, we varied the Elastic Plateau in the output space to see if A correlated. In both scenarios, we used the interactive callouts to simply drag the variable A or the part of the curve in the output callout that represented the Elastic Plateau.

Figure 6 shows the results of the interactive callouts from both directions. The visual confirms the relationship between the parameter A and the Elastic Plateau feature. As A increases the Elastic Plateau also increases, which shows that the visualization is useful in confirming known information about the specific shock physics ensemble. The sensitivity analysis for these two variables took less than two minutes with our approach.

7.2.2 Domain Expert Evaluation

We evaluated our visualization technique based on feedback from three domain experts, who were familiar with the shock physics application described above. One expert's background was in fluid simulation and

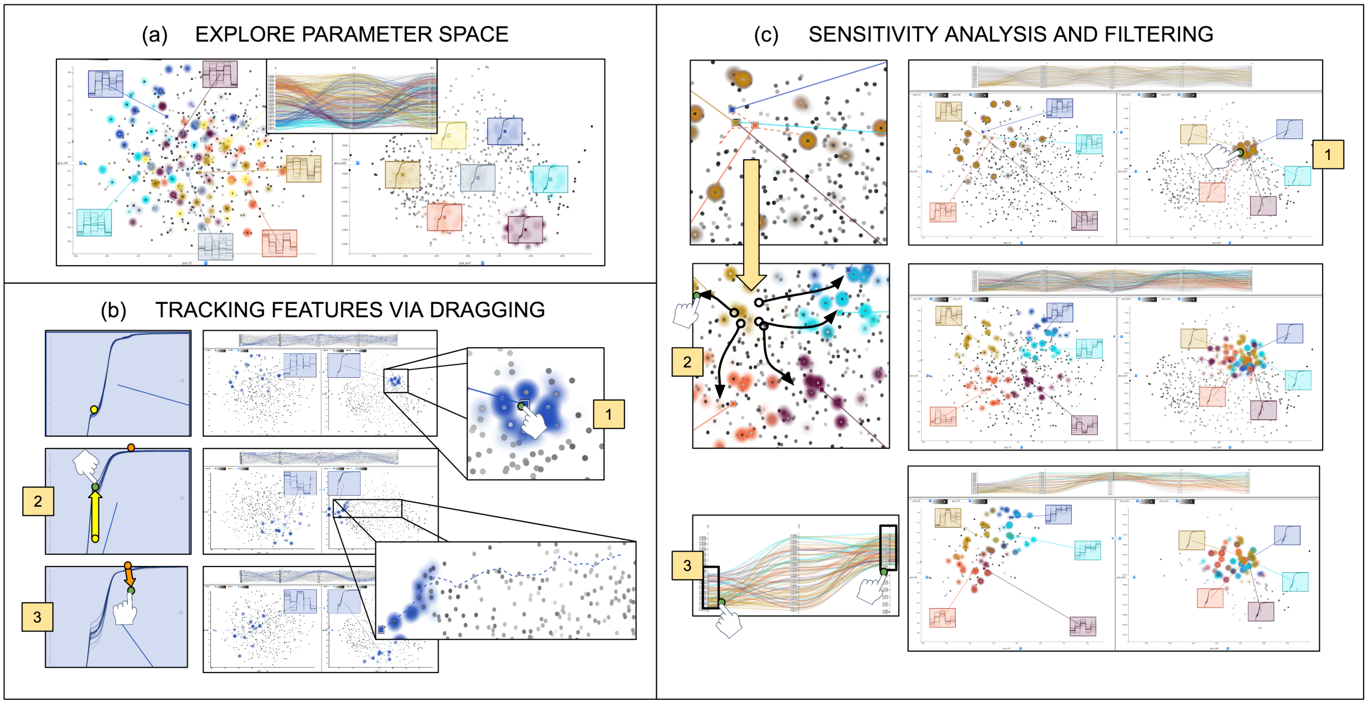

Fig. 5. Users interact with the system by (a) exploring the parameter space, (b) understanding features through direct manipulation, and (c) performing sensitivity analysis with filtering.

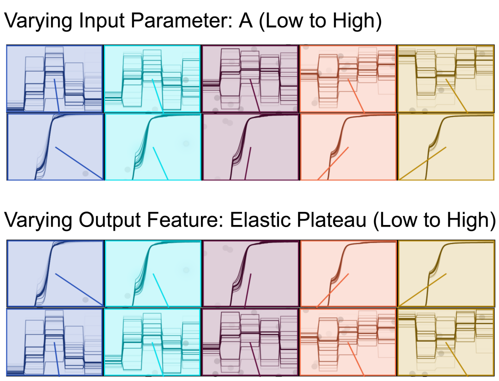

Fig. 6. Visual confirmation of the correlation between input parameter A and the Elastic Plateau output feature discovered in a sensitivity analysis. Top: As A increases, the Elastic Plateau increases when directly manipulating multiple callouts in the input space. Bottom: As the Elastic Plateau decreases, the input decreases when directly manipulating callouts from the output space.

model validation, and the two other experts were material scientists who ran shock physics experiments. We used a similar task work-flow as described in Section 7.1 to walk the users through our approach.

The current visual analysis tasks for material scientists include many of the techniques described in Section 2.2. These include parallel coordinate plots, dimensionally reduced sensitivity analysis scatter plots, and superimposed 1D velocimetry curves. The subject matter experts agreed that working in dimensionally reduced spaces like principal components analysis (PCA) is abstract and hard to relate points positioned in space with actual values.

Overall the feedback was positive, and the users were excited at the possibility of using tools like ours on new data sets, especially ones that have not been explored. The ease of navigating a space by "dragging and seeing" the relationships between the input and output immediately gave our approach utility. The users explained that it is useful to compare the spread of the tracked paths as they move through both input and output spaces. For example, a path that moves through one PCA space an appreciable amount and shorter in another intuitively describes the model's parameter relationships.

The actual path, itself, was a little abstract to understand its meaning, however, the users suggested that perhaps showing how the attributes change as the path progresses would be useful in relating it to actual values. They thought that it would useful to be able to see which parameters changed along the path while dragging a feature like the elastic plateau, perhaps in another linked view that represented a timeline or while hovering over the path.

The highlights were considered helpful in selecting similar data instances as a way of validating the uncertainty and sensitivity in the model. Again, the users suggested that a tight fit in one space with a tight fit in another, shows that the spaces correlate well, while a more scattered selection in the PCA space would show that the neighbors are not that similar from the other projection. The users liked the ability to see multiple highlighted sets from several different places on the dimensionally reduced views. This allowed them to compare the sensitivities across different criteria.

The users thought that being able to filter the ensemble from the parallel coordinates plot was vital and worked nicely in our application. They suggested that sometimes, especially with domain knowledge or intuition, there are sets of parameter values they would specifically like to study. For example, occasionally one parameter needs to be physically tight while observing how the rest of the parameters respond to the particular variable. This is one part of our application we added based on initial expert feedback, but it naturally fit into the work-flow, enabling new perspectives on the data.

The domain experts finally made two suggestions on where this application would be most useful in their current study. 1) They thought it would be great for aluminum optimization, where there is an unknown material, but we don't know the parameters that go into it. These are good approaches for optimizing a new data set. 2) They were excited about discussing how our approach had potential to solve a current inverse problem that exists in the shock physics community. Many scientists have parameterized the model described above, but most of what exists are other parameterized models. Model parameters can be compared from experimental data sets, but there is not a way to compare outputs. Our technique could be used to evaluate their models quickly. The users theorized "playing the what if game" by superimposing an experimental curve on top of the interactive output callout. Then the output could be modified to try to match the static experimental curve. For example, how close you can a user get the elastic plateau to match without the timing being off? If a model did not match the experiment then something interesting was found and the model might need to be modified. The users said our approach could potentially take the ambiguity out of this process.

8 Application to Synthetic Models

We also used our implementation to explore four other synthetic models using multidimensional scaling (MDS). These were simpler examples including a parameterized Gaussian, cosine, Catmull-Rom spline, and a Bzier curve. To increase the number of parameters in the Gaussian and cosine, we added a few output parameters like height and position offset. The splines had six input values spread across the x-axis with different heights. We were able to confirm the parameter relationships in both spaces. For Gaussian example, when we moved the mean position up on the output curve and pulled a point that was a short distance from the mean down, the curve was symmetric and the standard deviation went down in the input space as expected. For the spline examples, when we moved a control point up or down, the spline updated as expected. Similarly, when we moved a position on the spline up or down, the control point also moved accordingly. Finally, we were able to use the same input for both the Catmull-Rom spline and the Bzier curve, which allowed us to set the input callout to use the Bzier curve and the output callout to use the Catmull-Rom spline. The effect allowed us to use the Bzier curve to update the Catmull-Rom spline and the input parameters.

9 Discussion and Future Work

One of the key contributions of Drag and Track is to start a discussion about the implications of combining approaches that seem to be on opposite sides of a spectrum. We ask the questions, "What would an application look like that is both local-to-global and global-to-local?" or "What if output and input parameters searched the space the same way?" With Drag and Track, we are looking at the full data set and only a few instances. The direct manipulation of input and output parameters gives us both continuous (estimates) and discrete (nearest neighbors) results.

This flexibility, however, also makes our technique a bit difficult to categorize, causing confusing questions like "What is it?" We suggest that, instead of being a kitchen sink, the merger of the navigational strategies and the input and output domains helps us solve problems we might not have been able to otherwise. Our work is motivated by other visualizations like Design by Dragging [14], an interface that starts with one instance and allows users to search a space by directly manipulating it. One of the major drawbacks for Design by Dragging is the fact that users do not get a feel of how their search relates to the overarching space. In order to solve this problem, we immerse our directly manipulated queries inside DR views of the data, so there is never a question of where the query is. According to the user feedback, this is important for shock physics, where a single instance or estimate should tell us information about the space as a whole.

Although the global-to-local solution described above is useful, we also validated that our approach projects to the local-to-global approach. As in Berger et al [3], we calculate estimates with respect to an ensemble, but we allow for multiple queries. Our analysis in Section 7.2.1, suggests that the local-to-global approach here allows us to quickly visually compare multiple estimates along with their uncertainty distribution to confirm sensitivity information. The main thing to consider here is that the same interface and technique simultaneously solves problems that are global-to-local and local-to-global.

Estimation methods still suffer from the approximation technique used by the system. There is always the question of whether an estimate should be trusted. Although we use the inverse weighted distance for our implementation, Drag and Track allows for any other method that may more adequately match the data. Since our focus is on using estimators to annotate and track changes in the parameter spaces, estimates are naturally qualified by a nearest neighbor distribution. Therefore, Drag and Track could also be used for qualitative estimator analysis or validation. The tightness of the visual relationships with the nearest neighbors provides estimator model sensitivity information.

One major limitation of Drag and Track is the ambiguity between a VDI's predicted position on the DR spaces and the user's interaction described in Section 4.1. When a user clicks on a DR space, a VDI is created and the query set contains the x and y values of the clicked position. After the estimation step, the predicted data instance is reprojected back to the DR space, however, this point is not guaranteed to be the same place the user clicked since we rely on an estimation method and not the details of the DR method. One proposed solution is for the estimation method to be the inverse of the DR method, which works fine for PCA, but not other DR methods that do not have a clear inverse like MDS or simple scatter plots based on two single parameters. We originally tried to display both the query point and the predicted point to represent both desire and actual result, however this approach confused the users. Our solution was to show the query point while a user is dragging in the DR space and then show an interpolated predicted point for all other interactions. Since Drag and Track is more interested in the nearest neighbor distribution produced by the VDI, users do not appear to be confused or concerned with the location of the VDI except as a way to navigate a DR space.

A more robust solution to this problem would be to combine results from the estimator while also using properties of the DR method so that the two points agree. Cavallo and Demiralp [13] use optimization to minimize the changes in an instance, while Dis-Function [8] looks at how to change the distance function used by the DR to affect the position of points. iLAMP [15] uses a backwards projection method based on local neighborhoods. Endert et al [22] propose methods for changing the DR methods like MDS to optimize user's intent. Both the similarity search step and estimation step could use these types of methods to better approach changes related to the DR space.

Drag and Track is naturally built to use similar semantic interaction techniques [18, 19, 21] to change the DR projection method based on user intent, which a few tools implement [8, 20, 47]. In our case, the virtual data instances would work the same, however, users could simply move the ensemble data points on the DR views to change the DR method. Our work adds the ability to inform which changes need to be made by providing ensemble highlights of those points. If after searching in the output space, a user sees that the input parameters are spread out, he or she can simply move those points together to affect the DR view of interest. It would be interesting to explore how the two types of interaction could improve sensitivity and uncertainty analysis.

10 Conclusion

In this paper we present Drag and Track, an ensemble visualization technique that combines the power of multiple linked dimensionally reduced views with direct manipulation. We answer the call to action to both understand complex non-linear systems like applications in shock physics and aid in making high-dimensional projections intuitive. Our solution uses user-defined virtual data instances to annotate a parameter space via searching for similar instances and estimating parameter values. Our evaluation shows that the speed and ease of using such a system provides opportunities for the next generation of experimental physics applications and works toward solving difficult visualization challenges in understanding high-dimensional spaces. The hope is that this work inspires more organic interfaces that use direct manipulation to naturally study spaces beyond our current reach of knowledge.

11 Acknowledgements

The authors thank David Walters, Richard Sandberg, and Cindy Bolme for valuable feedback on Drag and Track as it relates to shock physics. Thanks to the artist Francesca Samsel for working with us to improve the colors for our visuals in Drag and Track. This work is published under LANL LA-UR-18-22844.

References

[1] J. Ahrens, S. Jourdain, P. O'Leary, J. Patchett, D. H. Rogers, and M. Petersen. An image-based approach to extreme scale in situ visualization and analysis. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 424–434. IEEE Press, 2014.

[2] D. Banesh, J. A. Schoonover, J. P. Ahrens, and B. Hamann. Extracting, visualizing and tracking mesoscale ocean eddies in two-dimensional image sequences using contours and moments.

[3] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. In *Computer Graphics Forum*, vol. 30, pp. 911–920. Wiley Online Library, 2011.

[4] A. Biswas, S. Dutta, H.-W. Shen, and J. Woodring. An information-aware framework for exploring multivariate data sets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2683–2692, 2013.

[5] A. Biswas, G. Lin, X. Liu, and H.-W. Shen. Visualization of time-varying weather ensembles across multiple resolutions. *IEEE transactions on visualization and computer graphics*, 23(1):841–850, 2017.

[6] C. Bolme, A. Mackinnon, and S. Glenzer. Fourth user workshop on high-power lasers at the linac coherent light source, 2017.

[7] J. Boteler and D. Dandekar. Dynamic response of two strain-hardened aluminum alloys. *Journal of applied physics*, 100(5):054902, 2006.

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

[9] S. Bruckner and T. Moller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1468–1476, 2010.

[10] D. E. Burton. FLAG, a Multi-Dimensional, Multiple Mesh, Adaptive Free-Lagrange, Hydrodynamics Code. In *Nuclear Explosives Code Developers Conference*, 1992.

[11] D. E. Burton. Lagrangian Hydrodynamics in the FLAG Code. In *Symposium on Advanced Numerical Methods for Lagrangian Hydrodynamics*. Los Alamos, NM, USA, LA-UR-07-7547, 2007.

[12] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen. The Construction of Compatible Hydrodynamics Algorithms Utilizing Conservation of Total Energy. *Journal of Computational Physics*, 146:227–262, 1998.

[13] M. Cavallo and Ç. Demiralp. A visual interaction framework for dimensionality reduction based data exploration. *ACM Human Factors in Computing Systems (CHI)*, 2018.

[14] D. Coffey, C.-L. Lin, A. G. Erdman, and D. F. Keefe. Design by dragging: An interface for creative forward and inverse design with simulation ensembles. *IEEE transactions on visualization and computer graphics*, 19(12):2783–2791, 2013.

[15] 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 *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pp. 53–62. IEEE, 2012.

[16] P. Dragicevic, G. Ramos, J. Bibliowitcz, D. Nowrouzezahrai, R. Balakrishnan, and K. Singh. Video browsing by direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 237–246. ACM, 2008.

[17] S. Dutta and H.-W. Shen. Distribution driven extraction and tracking of features for time-varying data analysis. *IEEE transactions on visualization and computer graphics*, 22(1):837–846, 2016.

[18] A. Endert, L. Bradel, and C. North. Beyond control panels: Direct manipulation for visual analytics. *IEEE computer graphics and applications*, 33(4):6–13, 2013.

[19] A. Endert, R. Chang, C. North, and M. Zhou. Semantic interaction: Coupling cognition and computation through usable interactive analytics. *IEEE Computer Graphics and Applications*, 35(4):94–99, 2015.

[20] A. Endert, P. Fiaux, and C. North. Unifying the sensemaking loop with semantic interaction. In *IEEE Workshop on Interactive Visual Text Analytics for Decision Making at VisWeek 2011*, 2011.

[21] 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, 2012.

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

[23] M. Enenhofer. Spatial-query-by-sketch. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pp. 60–67. IEEE, 1996.

[24] F. Ferstl, K. Bürger, and R. Westermann. Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):767–776, 2016.

[25] M. Hummel, H. Obermaier, C. Garth, and K. I. Joy. Comparative visual analysis of lagrangian transport in cfd ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2743–2752, 2013.

[26] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. ipca: An interactive system for pca-based visual analytics. In *Computer Graphics Forum*, vol. 28, pp. 767–774. Wiley Online Library, 2009.

[27] G. R. Johnson and W. H. Cook. A Constitutive Model and Data for Metals Subjected to Large Strains, High Strain Rates and High Temperatures. In *Seventh International Symposium on Ballistics*, pp. 541–547. The Hague, The Netherlands, apr 1983.

[28] A. Kumpf, B. Tost, M. Baumgart, M. Riemer, R. Westermann, and M. Rautenhaus. Visualizing confidence in cluster-based ensemble weather forecast analyses. *IEEE transactions on visualization and computer graphics*, 24(1):109–119, 2018.

[29] L. Liu, A. P. Boone, I. T. Ruginski, L. Padilla, M. Hegarty, S. H. Creem-Regehr, W. B. Thompson, C. Yuksel, and D. H. House. Uncertainty visualization by representative sampling from prediction ensembles. *IEEE transactions on visualization and computer graphics*, 23(9):2165–2178, 2017.

[30] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. In *Proc. Eurographics Conf. Visualization*, pp. 20151115–127, 2015.

[31] A. Madsen. Conceptual design report: Scientific instrument mid. (XFEL.EU TR-2011-008), 2011.

[32] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 389–400. ACM Press/Addison-Wesley Publishing Co., 1997.

[33] M. Mirzargar, R. T. Whitaker, and R. M. Kirby. Curve boxplot: Generalization of boxplot for ensembles of curves. *IEEE transactions on visualization and computer graphics*, 20(12):2654–2663, 2014.

[34] T. Ngo, D. Cutrell, J. Dana, B. Donald, L. Loeb, and S. Zhu. Accessible animation and customizable graphics via simplicial configuration modeling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 403–410. ACM Press/Addison-Wesley Publishing Co., 2000.

[35] H. Obermaier, K. Bensema, and K. I. Joy. Visual trends analysis in time-varying ensembles. *IEEE transactions on visualization and computer graphics*, 22(10):2331–2342, 2016.

[36] H. Obermaier and K. I. Joy. Future challenges for ensemble visualization. *IEEE Computer Graphics and Applications*, 34(3):8–11, 2014.

[37] S. Pajer, M. Streit, T. Torsney-Weir, F. Spechtenhauser, T. Möller, and H. Piringer. Weightlifter: Visual weight space exploration for multi-criteria decision making. *IEEE transactions on visualization and computer graphics*, 23(1):611–620, 2017.

[38] J. Patchett, F. Samsel, K. Tsai, G. R. Gisler, D. H. Rogers, G. D. Abram, and T. L. Turton. Visualization and analysis of threats from asteroid ocean impacts. Supercomputing, 2016.

[39] F. V. Paulovich, D. M. Eler, J. Poco, C. P. Botha, R. Minghim, and L. G. Nonato. Piece wise laplacian-based projection for interactive data exploration and organization. In *Computer Graphics Forum*, vol. 30, pp. 1091–1100. Wiley Online Library, 2011.

[40] H. Piringer, W. Berger, and J. Krasser. Hypermoval: Interactive visual validation of regression models for real-time simulation. In *Computer Graphics Forum*, vol. 29, pp. 983–992. Wiley Online Library, 2010.

[41] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pp. 233–240. IEEE, 2009.

[42] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE transactions on visualization and computer graphics*, 23(1):241–250, 2017.

[43] B. Saket, H. Kim, E. T. Brown, and A. Endert. Visualization by demonstration: An interaction paradigm for visual data exploration. *IEEE trans-*

actions on visualization and computer graphics, 23(1):331–340, 2017.

[44] A. Schulz, J. Xu, B. Zhu, C. Zheng, E. Grinspun, and W. Matusik. Interactive design space exploration and optimization for cad models. *ACM Transactions on Graphics (TOG)*, 36(4):157, 2017.

[45] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014.

[46] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pp. 517–524. ACM, 1968.

[47] J. Soo Yi, R. Melton, J. Stasko, and J. A. Jacko. Dust & magnet: multivariate information visualization using a magnet metaphor. *Information visualization*, 4(4):239–256, 2005.

[48] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer graphics forum*, vol. 33, pp. 401–410. Wiley Online Library, 2014.

[49] A. Tarantola. *Inverse problem theory and methods for model parameter estimation*, vol. 89. siam, 2005.

[50] T. Torsney-Weir, A. Saad, T. Moller, H.-C. Hege, B. Weber, J.-M. Verbavatz, and S. Bergner. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1892–1901, 2011.

[51] L. Van Der Maaten, E. Postma, and J. Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71, 2009.

[52] B. Wang and K. Mueller. The subspace voyager: Exploring high-dimensional data along a continuum of salient 3d subspaces. *IEEE transactions on visualization and computer graphics*, 24(2):1204–1222, 2018.

[53] J. Wang, X. Liu, H.-W. Shen, and G. Lin. Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots. *IEEE transactions on visualization and computer graphics*, 23(1):81–90, 2017.

[54] J. Waser, R. Fuchs, H. Ribicic, B. Schindler, G. Bloschl, and E. Groller. World lines. *IEEE transactions on visualization and computer graphics*, 16(6):1458–1467, 2010.

[55] M. E. Yumer, S. Chaudhuri, J. K. Hodgins, and L. B. Kara. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)*, 34(4):86, 2015.

[56] U. Zastrau, M. McMahon, K. Appel, C. Baehtz, E. Brambrink, R. Briggs, T. Butcher, B. Cauble, B. Chen, H. Damker, C. Deiter, J. Eggert, K. Falk, L. Fletcher, S. Glenzer, S. Gde, M. Harmand, A. Higginbotham, Z. Konpkov, D. Kraus, H.-P. Liermann, M. Nakatsutsumi, P. A., G. Priebe, R. Redmer, A. Schropp, R. Smith, P. Sperling, I. Thorpe, and S. Toleikis. Conceptual design report: Dynamic laser compression experiments at the hed instrument of european xfel. (REPORT-2017-004. XFEL.EU TR-2017-001), 2017.