Bridging Computation and Visual Communication of Change using Levels of Abstraction

Aritra Dasgupta *
Pacific Northwest National Laboratory
New Jersey Institute of Technology

Meg Pirrung †
Pacific Northwest National Lboratory.

Joe Bruce ‡
Pacific Northwest National Lboratory.

Jean Scholtz §
Pacific Northwest National Lboratory.

Kyungsik Han ¶
Ajou University.

Dustin Arendt ‖
Pacific Northwest National Lboratory.

Abstract

In time-varying data environments, computational methods like data mining machine learning are often used to automatically detect salient changes. For human monitoring and decision-making, these computationally detected changes need to be effectively communicated to an analyst. Currently, we lack a framework of visual communication that will allow us to design and evaluate alternative visualizations for communicating changes. In this paper, we contribute an abstraction based framework for designing communication-oriented visualizations to help analysts detect changes in real-time, with considerations for trade-offs between accuracy and efficiency of user tasks. We provide empirical evidence and case studies about the applicability of this framework in practice.

1 Introduction

Understanding change is a fundamental human task in a time-varying data environment (e.g., stock market prediction, cyber threat detection). While computational methods can be used to help detect important changes, a key challenge is how to communicate these changes to an analysts so that they can take actions in a timely manner. Changes are often too fast to notice, too many to remember, and too complex to understand. From a visual communication perspective, a key trade-off to address is that between accuracy and efficiency. When computational methods like a machine learning classifier is used to detect changes, it is important to convey all information related to the model output for analysts to gauge the degree of change accurately. However, for real-time tasks, if human attention is fixated on complex pieces of information that need a significant amount of effort for understanding, analysts could be less efficient in their decisions and miss important changes. Besides the information encoding aspect, there is also the effect of the human mental model on how the information is processed and perceived. Analysts' mental models depend on their training and background, and they have different requirements from a visualization. For example, a data scientist who wants to evaluate her models would need a detailed representation of the model uncertainty and the related parameters, while a domain expert, not typically trained in computational methods (e.g., a journalist), would need a more abstract and simpler change representation for understanding its meaning and context.

*e-mail: aritra.dasgupta@njit.edu
†e-mail: meg.pirrung@pnnl.gov
‡e-mail: joseph.bruce@pnnl.gov
§e-mail: jean.scholtz@pnnl.gov
¶e-mail: kyungsikhan@ajou.ac.k
‖e-mail: dustin.arendt@pnnl.gov

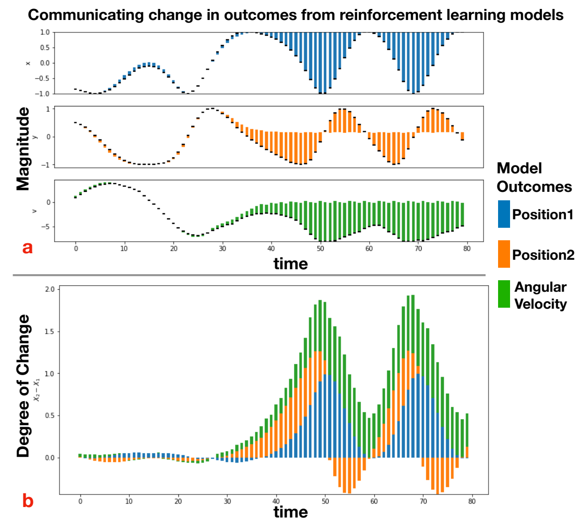

Figure 1: **An illustrative example for showing how different visual encodings influence communication of change.** Outcomes from a reinforcement learning model for solving the inverted pendulum problem is mapped to the bar charts. In a) each bar represents a magnitude at a particular time step, while in b) each bar directly encodes a change in magnitude. In the latter case, the information about absolute values of the parameters is lost at the cost of speeding up change detection.

This leads to a key research question: *how do we bridge the gap between computation and visual communication of changes for systematically addressing the design trade-offs?* To address this question, in this paper, we adapt and extend the visual abstraction based framework proposed by Viola and Isenberg, in the context of scientific visualization [16] , to the scenario where *a computational method is used to detect salient changes and the information about these changes are communicated visually to an analyst.*

We ground the abstraction framework in Shannon's theory of information communication where information is defined as a measure of the decrease of uncertainty for the receiver of a message [15]. If visualization is conceptualized as an information channel, then it needs to account for uncertainty with respect to how much information is encoded and how that information should be decoded in the perceptual and cognitive mental space of the analyst. These have been termed as encoding and decoding uncertainty respectively [7]. By instantiating different levels of abstraction with varying levels of encoding and decoding uncertainty, we are able to systematically evaluate trade-offs while designing visualizations for change communication. We have two main contributions in this paper. First, we present an abstraction-based framework in the context of visual communication of computationally detected changes in time-varying data. Second, we report on a user study with experts and non-experts for evaluating the effects of the different levels of abstraction on the accuracy and efficiency of human change detection tasks.

2 An Abstraction-Based Framework

Abstraction has been a demonstrated framework for deconstructing the nature of communication among humans [12]. Data abstraction has also been a successful concept for bridging the gap between system-level behavior and programmer's mental model [9]. By mapping the concept of abstraction to data visualization, we demonstrate that visual representations of data and results of automated methods can be instantiated and evaluated based on different levels of abstraction. Level of abstraction framework attempts to capture the necessary and sufficient amount of information that can be communicated through a visualization along a data-knowledge continuum [4]. With complex data, such representational primacy is a limiting notion [1] as in the case of discovery-oriented or high-level sensemaking tasks data often needs to be modeled in different ways, and we lack an objective measure of what to present to the user. Furthermore, well known examples such as the London Underground map [10] clearly demonstrate occasions where a more abstract and distorted representation of the underlying data can improve usability and decision making.

Recently, Viola and Isenberg proposed a theoretical framework on levels of abstraction for illustrative visualizations [16]. We adapt that framework in the context of visual communication of change, specifically with the aid of machine learning model predictions. The central tenets of our framework are the following: i) An abstraction-based framework should be able to communicate the uncertainty associated with predictions about change, and ii) The highest level of abstraction will enable users to notice changes more efficiently while the lowest level of abstraction will enable users to understand the degree and significance of changes more accurately.

2.1 Complexity of Change Communication

We consider deep reinforcement learning applied to the inverted pendulum problem to illustrate how visual communication of change can aid analysts in understanding model performance (Figure 1). The inverted pendulum is a classic control problem where the goal is to balance a rigid pendulum in its inverted, i.e., unstable, configuration. The input to the system is a small torque, which changes the angular velocity of the pendulum. The state of the system is characterized by the x and y coordinate of the pendulum, and its angular velocity, v. The deep reinforcement learning algorithm attempts to learn the optimal control policy that maximizes the time the pendulum spends in the inverted position while minimizing the amount of torque used.

We use this example to show how different choices of change encoding can affect an analysts' perception. In case of the inverted pendulum problem, the learning process is segmented into episodes, where the model attempts to solve the inverted pendulum problem starting with random initial conditions. Each episode is a sequence of state action pairs, where the state is the configuration of the pendulum, and the action is determined by the model and results in the next state of the pendulum. Because the state of the pendulum is described by an (x, y, v) tuple over time, the state sequence is a multivariate time series.

The task of an analyst is to compare two episodes which represent two multivariate time series and diagnose time points when there is a divergence of performance across the episodes. This translates to a pairwise visual comparison problem, where we can vary how change is encoded by varying the degree of explicit encoding [11]. In the first case (Figure 1a), a small multiple of bar charts is used to show the difference between the values, as well as the actual values over time. For the x variable, a bar is drawn between the value of x_1 and x_2 at each time step 1 and 2 refer to the first and second episode. The small black mark in each bar indicates the end of the bar that corresponds to X1. Here, change is implicitly encoded. The same technique is also used for y and v in separate multiples. The second bar chart (Figure 1b) preserves the relative sizes of differences, but we lose the context of the change, i.e., for what values of x, y, and v it occurs. Area drawn below zero indicate that a value for that variable was larger for episode 1 compared to episode 2.

Comparing thee episodes allows us to see when the first episode's behavior deviated from the second. For $t < 30$ both episodes appear nearly identical. However, between $t = 30$ and $t = 40$ a deviation has occurred – the second system reaches a rest state whereas it appears that the first overshoots, causing an oscillation that is not recovered from. In the second bar chart, since change is more directly encoded, analysts will be able to spot big or small changes faster than they might be able to using the first bar chart. However, the latter is more accurate in providing more context to the degree of the changes.

This simple example shows that by controlling explicit encoding, we can communicate change to an analyst. However, we argue that for communicating more complex changes, merely accounting for explicit encoding is not sufficient. This is owing to the following three reasons. First, in a general scenario, comparisons have to be performed across many time series and not just a pair of time series (which are episodes in this case). In such cases, a fully explicit encoding of change is not possible. Second, the amount of information that needs to be encoded about a model prediction might be dependent on a user preference and might not be known *a priori*, unlike the inverted pendulum case. Third, there can be other perceptual factors like clutter or overlap in the visualization, which can inhibit how clearly visual structures get communicated to a user. All of these factors necessitate that we have an end-to-end understanding of how model predictions get translated into visual channels and ultimately, how these patterns are decoded by a user. This is the motivation behind our abstraction-based framework, which we discuss next.

2.2 Visual Uncertainty

Similar to a communication channel, visualization involves encoding and decoding of information as data gets progressively transformed in the course of visual mapping, and the subsequent stages of human perception and cognition. Visual uncertainty has been defined as the the uncertainty that is associated with a visualization during encoding (in the screen-space) and decoding of information (in the mental space of the user) [7, 8]. We argue that different degrees of encoding and decoding uncertainty, when combined in the context of an analytical task, can lead to different levels of abstraction while communicating change. For illustrating this concept, we discuss a more complex change detection scenario. We assume that the analyst has trained an ensemble of classification models to categorize a particular system, so there will be one binary classification model trained per class, and each model reports the probability that an observation belongs to its class. This results in a probability time series–one probability per model/class over time. Analysts need to interpret and compare across this multivariate time-series to determine when a significant change is occurring. In the following, we discuss the different sources of encoding and decoding uncertainty at the different stages of the visualization pipeline [5] that need to be considered for designing visualizations with the goal of communicating change (Figure 2).

Encoding Uncertainty: At the encoding stage, the first design decision involves the amount of information about a model prediction that is included as part of a visualization. Model prediction scores can be necessary but not sufficient in cases where analysts also want to know the model confidence behind each prediction. The exclusion of any relevant information will lead a *lack of completeness*, while at the same time, including less information will help analysts in spending less time on processing the information in a real-time change detection task. The second design decision for the visual mapping stage is which visual variables should be used for encoding change. For example, in the previous bar chart example, change was encoded

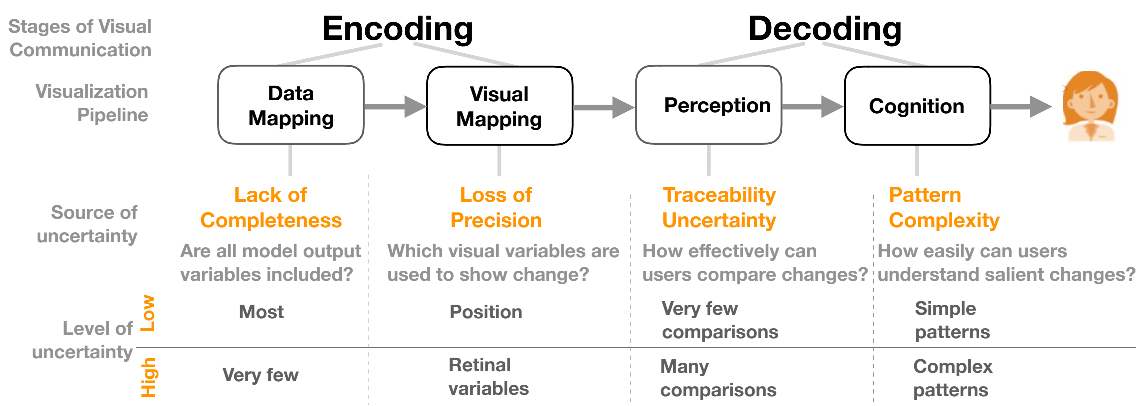

Figure 2: **Design considerations for visual communication of change involve different levels of encoding and decoding uncertainty at the multiple stages of data transformation along the visualization pipeline**. High and low levels of encoding and decoding uncertainty can be systematically combined to instantiate visualizations at different levels of abstraction.

with the position variable, which ensures high accuracy. Change can also be encoded through retinal variables like size, color etc., in case position is used to encode some other piece of information. While using color can sometimes help lead to a pop-out effect and focus our attention on salient changes [13], it is less accurate than position can lead to a *loss of precision* in gauging the magnitude of change.

Decoding Uncertainty: At the decoding stage, a user needs to first compare changes across multiple time-series to understand where salient changes have occurred. Tracing changes across multiple time-series can be impeded if a visualization is cluttered or if there are too many overlapping time-series, leading to *traceability uncertainty*. Finally, at the cognition stage, users should be able to form conclusion based on observation of the patters. In this context, where a classifier outcome is used as a way to compute information about changes, *pattern complexity* can be caused by the lack of familiarity of a user with machine learning, as the patterns might need domain expertise for proper interpretation.

2.3 Levels of Abstraction

We systematically combine the different amounts of encoding and decoding uncertainty for deciding a level of abstraction (LOA). Levels of abstraction can be compared for a particular type of visualization (e.g., bar chart, line chart, etc.), designed to solve a specific visual communication task (e.g., change detection). Our LOA framework is exemplified using a line chart example. Different variants of line charts can be used to instantiate visualizations at different levels of abstraction. The information we believe is relevant to decision making in this context is the model confidence (i.e., predicted probabilities), the relative ordering of the models in terms of confidence, and which models' confidence are significant (i.e., above an arbitrary threshold). The visualizations encode time on the horizontal axis, and as new data arrives, it is appended to the visualization (to the right) (Figure 3). Each colored line represents a class as predicted by the machine learning classifier. The thickness of the line at a given time corresponds to whether that prediction probability is currently above or below an arbitrary detection threshold – thick lines are above the threshold, thin lines are below.

High Level of Abstraction: At this level, the encoding uncertainty is maximum while the decoding uncertainty is minimum. We only encode the minimal amount of information relevant to the change detection task using a simple enough representation so that analysts can readily decode what/when are the most salient changes. This level aims to maximize efficiency while sacrificing accuracy. As shown in Figure 3, at the highest LOA, the only encoding is thickness, which shows when the model belief is above or below the detection threshold. This minimizes decoding uncertainty as one can directly compare the gray and blue classes and find the time step when either line is exclusively above the threshold. But potentially too much information is lost because we don't know which class is the most likely if two are simultaneously above the (as is the case with the blue and red classes). We also may have less ability to determine when the change has occurred – in the example it appears to have occurred between step 10 and 16, but we cannot refine this range with much certainty.

Low Level of Abstraction: The most straightforward and least abstract approach is to directly map the probabilities onto the vertical axis, so that more likely classes are higher – we consider this technique to have low abstraction. The thickness of the line encodes whether the prediction is above the detection threshold. Traceability uncertainty and pattern complexity increases here as one has to make multiple comparisons and there are lots of overlapping lines leading to clutter. This visualization is likely to contain more information than is necessary for making the determination that an important change has occurred. We can determine with some certainty that the change between the blue class and the gray class occurred around step 11. However, the complexity of the chart increases the difficulty of making the visual comparison despite decreasing the uncertainty.

In-between or Medium Level of Abstraction: A third alternative that lies between these two extremes is to encode the rank of each class on the vertical axis. This approach divides the vertical space into discrete levels and assigns the most likely model to the top level, the second most likely model to the second from the top level, etc. We consider this approach to be at a medium level of abstraction, as it aims to minimize the loss of information as well as the human effort to detect a salient change. Because relative rank changes less quickly than the model belief in this example, the charts appear less complex compared to the low level abstraction. We can quickly determine what is the most likely class at any given time by tracing the top row, which is always in the same position. We can also quickly find other candidate classes by considering the 2nd, 3rd, etc. rows that are above the detection threshold, because they are thicker.

By systematically instantiating visualizations along a level-of-abstraction continuum, analysts can quickly move up and down the ladder of abstraction and look at different data-driven perspectives without having to tune low-level parameters and components of the visualization process. An optimal level of abstraction is one where we need to lose a certain amount of information to ensure the visualization is expressive [2] about the key patterns and helps amplify human cognition [3]. This is also closely linked to the inherent uncertainty in a visual analytics system. Though researchers have discussed modeling uncertainty in the visual analytics system [6], the relation between such uncertainty and users' goals are often unclear. An abstraction based framework is parameterized by user-defined goals, and therefore makes an explicit connection between analytical uncertainty and the insight derived by the user.

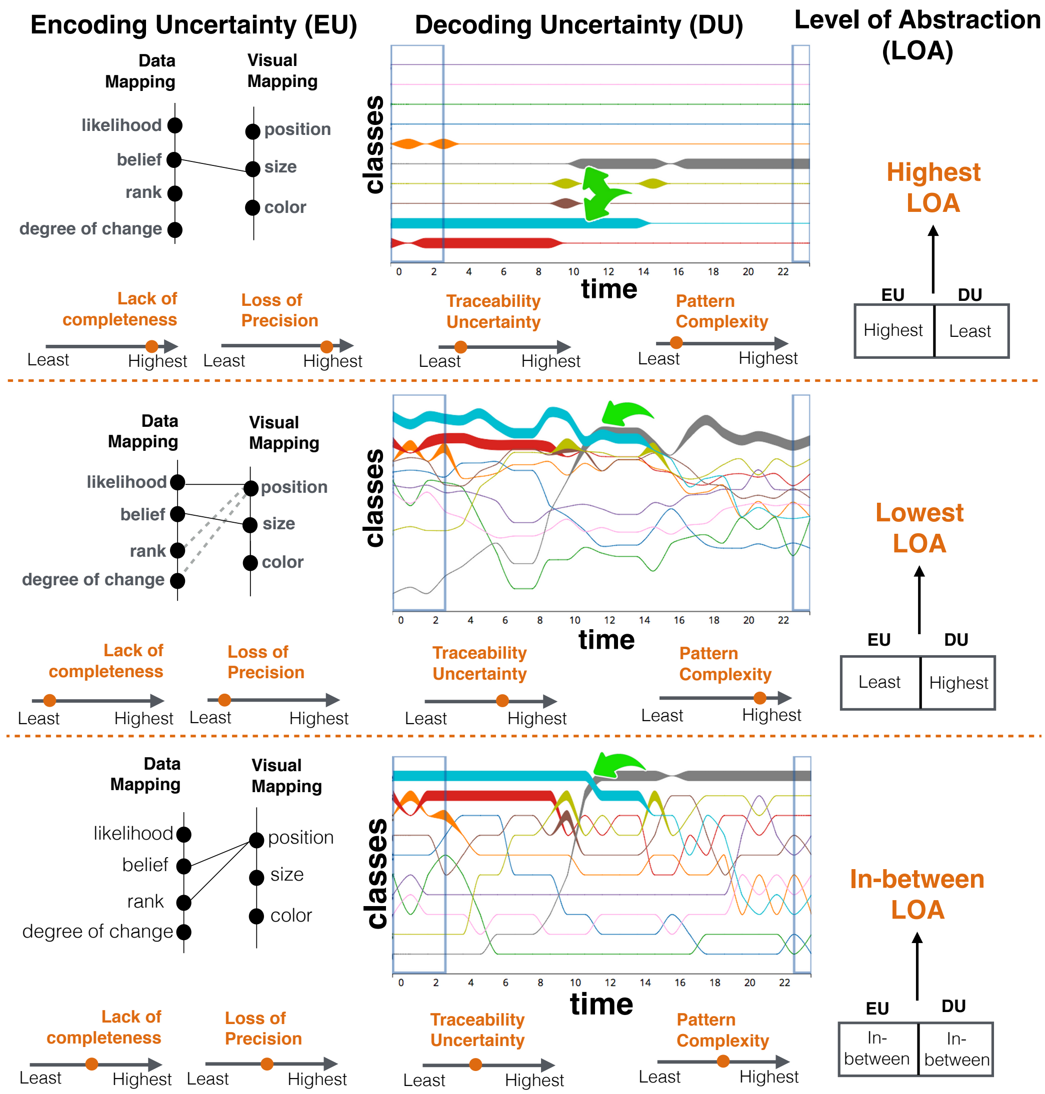

Figure 3: **Instantiating visualizations based on levels of abstraction with considerations for design trade-offs.** A hypothetical example where level of abstraction varies based on considerations for encoding and decoding uncertainty. Salient changes are indicated by green arrows. (a) In the least abstract view, the model belief and likelihood are directly encoded. (b) In the next view the relative ordering of models by belief is also encoded. (c) In the most abstract view, only whether a model's belief is above a threshold is encoded. A dotted line indicates information that is not directly encoded, but implicitly recoverable from the visualization. In high LOA, we only know what is changing, while in case of other abstraction levels, other change related information, like the degree of change or the most salient changes, can be inferred.

3 Empirical Evaluation

We conducted a within-subjects user study to understand the relationship between level of abstraction, user expertise, and user performance on a the change detection task we described previously. We recreate a time-varying data environment under controlled conditions for evaluating the effects of different levels of abstraction on users' performance. In this section, we describe the rationale behind the grouping of participants, task selection, metrics and hypotheses, and the data selection for our experiments. For this experiment we focus only on the task of change detection.

3.1 Study set up

We chose to employ synthetically generated time series data in order to a have sufficient level of experimental control to effectively test our hypotheses. We developed a biased random walk method that allowed us to take an arbitrary labeled multivariate dataset and generate a time series of predicted probabilities over the distinct classes in that dataset. The resulting time series exhibit a subtle change from one class to another. Our technique gives us enough control to specify between which pairs of classes the change occurs, and how quickly the change occurs. All stimuli that participants see are variations on this same theme, but have different class changes. For the study discussed in this paper we chose to start with the "Optical Recognition of Handwritten Digits Data Set" [18], available in the python scikit-learn package, which has 1797 samples (which are images of handwritten digits), 64 features (which are pixel intensity values ranging from 0 to 16), and 10 classes (digits zero through nine). We recruited participants through mailing lists and word-of-mouth. A total of 28 participants completed a one-hour user study. All participants work at a research laboratory, and their job titles range from research scientists, software engineers, statisticians, administrators to graduate interns. Based on their self-reported knowledge and experience in machine learning and visualization, 19 were assigned to the expert group and 9 were assigned to the non-expert group. We evaluated our visualizations with a high level task, which we described previously as the "change detection" scenario. In this scenario, a machine learning classifier has been trained to provide predicted probabilities across a set classes given some combination of features and values. This classifier has been deployed in a time-varying environment to summarize a system as it changes over time. The user's task is to monitor the time series of probabilities generated by the classifier, and report when a significant change has occurred. Often determining "significant change" will depend on the user's judgement and interpretation of all of the predicted probabilities, and is not as simple as picking the class with the largest probability at a given time. We show the user a visualization containing the first few time windows of predicted probabilities. We allow the user to advance time at their own pace. When time is advanced, the visualization is built incrementally by revealing data from the next time step to the right of the existing data. The user advances the visualization, revealing more time windows, until she detects a significant change. After the user detects a change, she indicates the time window in which the change occurred. The user may have to reveal several windows beyond the actual change event for the change to become visually salient. Allowing users to advance the time series themselves eliminates effects related to reaction time and change blindness that might occur with "live" data.

3.2 Metrics & Hypotheses

We used the following metrics to measure performance: *Correctness* measures whether the user entered the correct answer, within one time window of the ground truth. *Time on task* is the amount of time (wall clock) that the user spent during the task. *Accuracy* is the difference between the user's response (where she believes the change occurred) and the ground truth. *Efficiency* is the difference between the number of windows the user reveals, and the ground truth (the window where the change occurred) *Confidence (subjective)* is how confident the user feels that their answer is correct. *Understanding (subjective)* is how well the user understood the task.

Visualizations with different levels of abstraction will contain different amounts of encoded information. We expect the metrics to reveal performance differences across the level of abstraction. Each visualization contains sufficient information to solve the task, so the visualization with the highest level of abstraction contains the least amount of extraneous information. We expect that the lack of information in more abstract visualizations will be observable–we expect users to reveal more time windows before the change becomes obvious. Once enough windows are revealed, the visualization should contain enough information to accurately solve the task. We also expect this to lead users to be less confident in their answers.

The study consisted of three main sections. First, participants were asked to answer the questions about their demographics as well as their familiarity and experience in visualization and machine learning. Second, participants completed the main part of the study in which participants performed the change detection task with visualizations at the three different levels of abstraction. We used the visualizations in Figure 3 for the change detection task for this purpose. Lastly, participants rated the three different levels of abstraction with respect to confidence and understanding. In the main part of the study, each participant performed the change detection task 3 times with different datasets for each of the 3 different levels of abstraction. Thus, participants performed 9 tasks total with a different dataset used for each of the 9 tasks to prevent participants from memorizing the answer. Trials were blocked by level of abstraction, and the participant received training on how to interpret the particular visualization before each block. To counter-balance the experiment, the order of the blocks, and the order of the datasets within each block were randomized. We used time series between 10 and 50 steps in order to control the difficulty of the task.

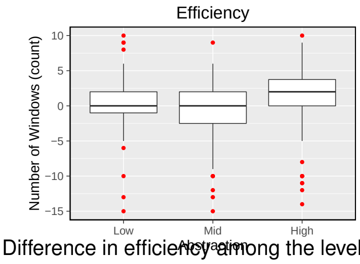

Figure 4: Difference in efficiency among the levels of abstraction. Low LOA showed higher efficiency (average: 0.144) than the Mid (-1.223) and the High LOA (2.045). Participants were less efficient with the most abstract visualization. Red dots indicate outliers.

4 Results

Because participants were measured more than once on the same dependent variable, we fit a mixed effects analysis of variance (ANOVA) model with a normal conditional distribution and random effects for repeated measures to account for the non-independent nature of the data [17]. A repeated measures ANOVA model carries the extra burden of an assumption of sphericity (i.e., that the relationship between pairs of experimental conditions is similar). In other words, while parametric tests based on the normal distribution assume that data points are independent, our data points for different conditions came from the same individuals. Therefore, a Mauchly's assumption of sphericity test was applied to determine whether the relationship between pairs of experimental conditions is similar. Using Manuchly's test, the null hypothesis is that the sphericity assumption is not violated (p-value is greater than 0.5) [14]. We had three experimental fixed variables: three levels of abstraction (Low, Mid, and High LOA), a between-subject factor of knowledge and experience in machine learning and visualization (MVIS–non-expert and expert groups), and three levels of order. Data were analyzed using a mixed effects two-way repeated measures ANOVA with a within-subjects factor of subscale. We measured the main and interaction effects of the three fixed variables on the four dependent variables (i.e., completion time, efficiency, accuracy, and the number of correct answers). A model was fit independently to each dependent variable. We used SPSS for the analysis. Among three levels of abstraction, significant differences were found in efficiency, accuracy, and the number of correct answers, as reported below.

Efficiency: According to our definition, better efficiency indicates that a participant stopped revealing time windows closer to the ground truth. We tested significant effects from the target variables on the response variable of efficacy, where the sphericity assumption was not found to be violated. As a result, no interaction effects were found, but one of the main effects, LOA, showed a significant effect ($F(2,24) = 6.15$, $p = 0.002$) as depicted in Figure 4. A post-hoc Linear Discriminant Analysis (LDA) test indicated that Mid LOA and High LOA differed significantly ($p = 0.003$) and Low LOA and High LOA differed marginally ($p = 0.08$). According to this result, efficiency decreased as the LOA increased. In general, the participants revealed more time windows than necessary by a very small amount, on average, for the Low LOA but revealed fewer windows than needed for the Mid LOA. On average, the participants revealed 2.5 extra windows for the High LOA, which means that the participants were less efficient with the most abstract visualization. Although there was no significant difference in efficiency between experts and non-experts ($p = 0.20$), the expert group generally revealed more windows beyond the ground truth (average: 1.1 windows), whereas the non-expert group average was reached before the window in which the ground truth change occurred (average: -0.5 windows).

Accuracy: There was no violation of sphericity assumption in measuring accuracy. The result showed significant variation in the interaction effects between Order and LOAs, $F(9,15.9) = 2.404$, $p = 0.05$. Figure 5 illustrates the difference in accuracy among three levels of abstractions over time. There was no significant difference among

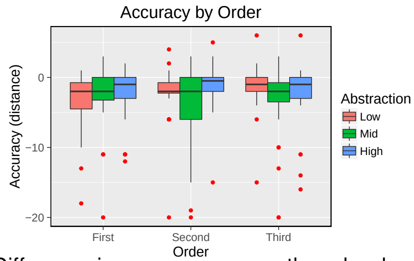

Figure 5: Difference in accuracy among three levels of abstraction. It seems that participants struggled the most with the Mid LOA for any time points beyond the first as the Mid LOA showed the lowest accuracy (average: -3.3 windows) compared to the Low (average: -1.8 windows) and High LOA (average: -1.2 windows). Interesting note is that the High LOA showed the highest accuracy even with the lowest efficiency (Figure 4). Red dots indicate outliers.

three levels of abstraction for tasks done first in each block. However, participants struggled most/had a further distance from the ground truth with the Mid LOA compared to both the High and Low LOA. If there was a learning effect then participants should understand the task better in the second and third task within each block. The difference in mean distance from the ground truth between the High LOA and Mid LOA were significantly different with the second task Order ($p = 0.04$) and between the Low LOA and Mid LOA were significantly different with the second and third task ($p = 0.08$ and 0.04, respectively). This implies the participants struggled the most with the Mid LOA for any time points beyond the first task.

Correct Answers: We measured the number of correct answers by the participants. The sphericity assumption was not violated, and ANOVA results showed significant interaction effects from Order*LOA. Overall, the High LOA showed the most correct answers (52%), followed by the Low LOA (45%) and Mid LOA (36%). A significant difference was found in the number of correct answers between the Low and Mid LOA with the third task in the block ($p = 0.035$). Participants in the Low LOA answered more correctly over time ($p = 0.05$ between 1st and 3rd Order). When we considered this result with the one from efficiency, interestingly, there was no significant correlation between efficiency and the number of correct answers. This indicates that higher efficiency does not necessarily lead to having more correct answers. This result was especially influenced by the High LOA as the participants revaled more windows (Figure 4) for the High LOA but instead answered more correctly.

5 Conclusion and Future Work

We developed an abstraction-based framework to assist with visual communication techniques for change detection. The key concept behind LOA is that information will be lost and uncertainty will be introduced when data is transformed to a visual representation. However, loss of information is not always harmful to the user – sometimes information loss can be beneficial when it highlights the most task relevant attributes of the data. While this is counterintuitive, sometimes "less is more", where decoding uncertainty is minimized at the cost of introducing high encoding uncertainty, thereby reducing the amount of information a user has to process for a real-time change detection. A limitation of the study may be the relatively small sample size. We caution that generalizations from this study are limited due to the relatively small sample size. While our experiment has focused on time-varying data, we believe our framework can be to be applied in other contexts and for different visual encodings. Such encodings could reasonably include scatter plots, parallel coordinates, stream graphs, etc.

We believe that more abstract visualizations are beneficial when the combination of data, task, and visual encoding require the user separate signal from noise in a visual manner. Certainly topological tasks such as route planning (i.e. shortest path) could be considered within the LOA framework, as the archetype LOA visualization is the metro map. Others could include visual search tasks such as anomaly detection and visual clustering, where the user must search for something interesting amidst many distractors. For these tasks, more abstract visualizations might simplify the view enough such that the task becomes pre-attentive, which we would expect to significantly improve user performance. The abstraction framework provides us with a systematic way to think about the mapping between outcomes from computational methods and communication-oriented visualizations, helping us design visualizations tailored to a particular task and degree of user expertise.

6 Acknowledgment

This work was funded by the Laboratory Directed Research and Development Program (LDRD) at PNNL. Battelle operates PNNL for the U.S. Department of Energy (DOE) under contract DE-AC05-76RLO01830.

References

[1] R. Amar, J. T. Stasko, et al. Knowledge precepts for design and evaluation of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):432–442, 2005.

[2] S. K. Card and J. Mackinlay. The structure of the information visualization design space. In *In Proc., Information Visualization*, pp. 92–99. IEEE, 1997.

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

[4] M. Chen, D. Ebert, H. Hagen, R. S. Laramee, R. Van Liere, K.-L. Ma, W. Ribarsky, G. Scheuermann, and D. Silver. Data, information, and knowledge in visualization. *Computer Graphics and Applications, IEEE*, 29(1):12–19, 2009.

[5] E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Information Visualization*, pp. 69–75. IEEE, 2000.

[6] C. D. Correa, Y.-H. Chan, and K.-L. Ma. A framework for uncertainty-aware visual analytics. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pp. 51–58. IEEE, 2009.

[7] A. Dasgupta, M. Chen, and R. Kosara. Conceptualizing visual uncertainty in parallel coordinates. In *Computer Graphics Forum*, vol. 31, pp. 1015–1024. Wiley Online Library, 2012.

[8] A. Dasgupta, J. Poco, Y. Wei, R. Cook, E. Bertini, and C. Silva. Bridging theory with practice: An exploratory study of visualization use and design for climate model comparison. *IEEE Transactions on Visualization and Computer Graphics,*, 21(9):996–1014, 2015.

[9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design patterns: Abstraction and reuse of object-oriented design. In *European Conference on Object-Oriented Programming*, pp. 406–431. Springer, 1993.

[10] K. Garland. *Mr Beck's Underground Map*. Capital Transport, 1994.

[11] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[12] S. I. Hayakawa. *Language in action: A guide to accurate thinking, reading and writing*. Harcourt, Brace and Company, 1941.

[13] C. Healey and J. Enns. Attention and visual memory in visualization and computer graphics. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1170–1188, 2012.

[14] J. W. Mauchly. Significance test for sphericity of a normal n-variate distribution. *The Annals of Mathematical Statistics*, 11(2):204–209, 1940.

[15] C. E. Shannon, W. Weaver, and A. W. Burks. The mathematical theory of communication. 1951.

[16] I. Viola and T. Isenberg. Pondering the concept of abstraction in (illustrative) visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(9):2573–2588, 2018.

[17] E. Vonesh and V. M. Chinchilli. *Linear and nonlinear models for the analysis of repeated measurements*. CRC press, 1996.

[18] L. Xu, A. Krzyżak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, man and cybernetics, IEEE transactions on*, 22(3):418–435, 1992.