# Supporting Co-Adaptive Machine Teaching through Human Concept Learning and Cognitive Theories

Simret Araya Gebreegziabher
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN, USA
sgebreeg@nd.edu

Yukun Yang
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN, USA
yyang35@nd.edu

Elena L. Glassman*
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA, USA
glassman@seas.harvard.edu

Toby Jia-Jun Li*
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN, USA
toby.j.li@nd.edu

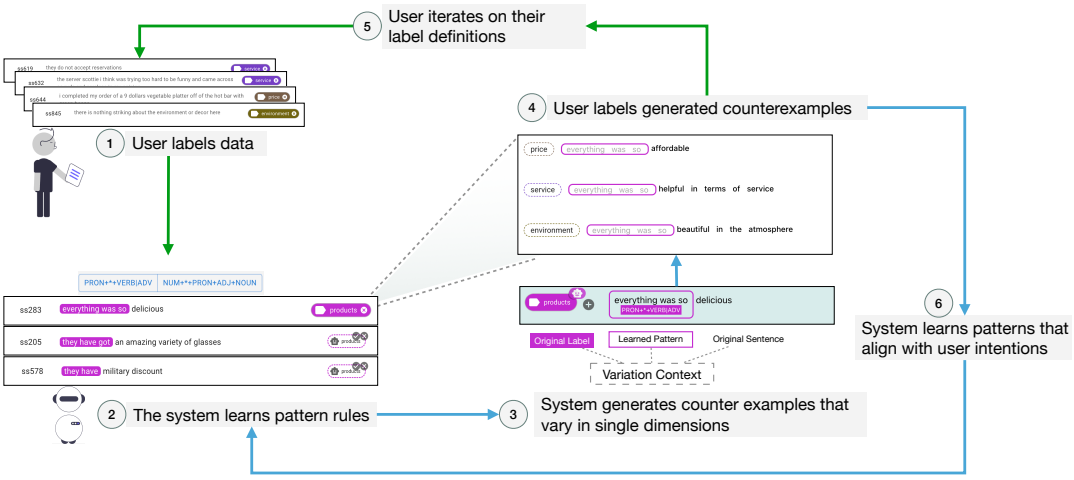


**Figure 1: A user is iteratively teaching a neuro-symbolic model to distinguish between different concepts (labels). The process begins with the user labeling some data points (1). This allows the neuro-symbolic model to learn pattern rules about the label and suggest annotations to unseen data points (2). As the user reads and accepts or rejects model-suggested labels, Mocha uses an LLM to generate counterfactual examples that structurally resemble the original data point and match the original patterns but have different predicted labels (3). When presenting the generated counterfactual examples to the user Mocha emphasizes the changed parts and de-emphasizes the carried-over parts of each sentence while highlighting (in the associated label's color) where the current neuro-symbolic model would fail on the generated counterfactuals (4). The user then assigns labels to the generated counterfactual examples by accepting or rejecting the LLM-generated labels to be used in consecutive model training (5). As the user provides feedback through labeled data, the model iteratively learns and adjusts its decision boundary, to better align with the user's mental model and labeling criteria (6).**

*Co-senior authors contributed equally.


*CHI '25, Yokohama, Japan*

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713708

## Abstract

An important challenge in interactive machine learning, particularly in subjective or ambiguous domains, is fostering bi-directional alignment between humans and models. Users teach models their concept definition through data labeling, while refining their own understandings throughout the process. To facilitate this, we introduce Mocha, an interactive machine learning tool informed

by two theories of human concept learning and cognition. First, it utilizes a neuro-symbolic pipeline to support Variation Theory-based counterfactual data generation. By asking users to annotate counterexamples that are syntactically and semantically similar to already-annotated data but predicted to have different labels, the system can learn more effectively while helping users understand the model and reflect on their own label definitions. Second, Mocha uses Structural Alignment Theory to present groups of counterexamples, helping users comprehend alignable differences between data items and annotate them in batch. We validated Mocha's effectiveness and usability through a lab study with 18 participants.

## CCS Concepts

• **Human-centered computing → Systems and tools for interaction design**; **User interface programming**.

## Keywords

human-AI collaboration, machine teaching, variation theory, structural alignment theory

**ACM Reference Format:**

Simret Araya Gebreegziabher, Yukun Yang, Elena L. Glassman, and Toby Jia-Jun Li. 2025. Supporting Co-Adaptive Machine Teaching through Human Concept Learning and Cognitive Theories. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3706598.3713708

## 1 Introduction

In supervised and semi-supervised machine learning (ML) pipelines, labeled data is a vital component of training and validating models [46]. Interactive ML (IML) methods, like active learning [3], continuously apply human feedback during model training to iteratively build and refine the model [35, 42, 43]. A targeted approach in IML is machine teaching (MT) [60], an interactive framework that allows users to devise and select useful data for labeling, with the goal of teaching the model relevant features during training [7, 18]. Through labeled data, human users "teach" an underlying concept to the model [50]. In a MT pipeline, humans act as experts on concepts with an explicit goal of creating ML models through a teacher-student interaction. The approach of MT heavily resembles the co-adaptivity of human-to-human teaching in which the teacher illustrates concepts, assesses the learner's progress and evolution of a concept [40, 56], and iteratively revises their teaching approach [38, 60].

Prior work in IML has incorporated human input into model training and refinement through iterative processes of labeling and reviewing [50]. A prominent form of input is when humans provide labels for representative training examples, with the goal of adjusting model parameters [53]. When training samples are scarce, model performance heavily depends on the quality of available training examples [15]. However, relying exclusively on existing examples is not ideal for tasks requiring nuanced understanding of user intentions, as these examples often fail to represent diverse and edge-case scenarios [31]. While using synthetic data for active learning has promising results to mitigate data scarcity [49], much of this work prioritizes optimizing model performance, offering limited support for human learning and critical reflection—an essential component of MT. To support users in building accurate conceptual models during model teaching, Gillies et al. [30] argue that interfaces should be reframed to account for human cognitive processes. Data labeling as a cognitive task—including defining a concept or determining how two similar objects may have different labels—requires both comparison and integration [62].

To address these needs, we introduce an interactive tool called Mocha. Mocha presents novel interaction mechanisms inspired by two human cognition theories. First, the Variation Theory of human concept learning [44] informed a new approach to generate synthetic counterfactual data for users to annotate. Secondly, the Structural Alignment Theory [27] guides the design of Mocha's interface for presenting generated counterfactual examples in batch, which assists users in perceiving and comprehending the alignable differences between data items and annotating these data items in batch.

**Counterfactual Data Generation.** In the counterfactual data generation phase, once a user annotates a small initial dataset, Mocha employs a Variation Theory (VT) [44]-based pipeline to create synthetic data. VT posits that human learning occurs when learners experience variation across critical and superficial aspects of a concept—through exposure to contrasting examples that systematically vary along different critical and superficial feature dimensions. Inspired by VT, our pipeline starts with the neuro-symbolic model's current (and potentially imperfect) learned pattern rules, which can be thought of as feature dimensions. It then generates counterfactual data that are syntactically and semantically similar enough to an already-annotated datum that they would be given the same label by the neuro-symbolic model's pattern rule, but different enough that they would be given a different label by a standard pre-trained large language model. Therefore, the generated data poses the hypothetical question [19]: "How should the model's prediction change if certain aspects of the input were altered?"

Consider this analogy to illustrate the counterfactual approach for refining concept boundaries; a user and a model are negotiating how to define a sandwich. Although both may start with their own definition, neither is accurate or specific. We suppose that the user starts with a definition *"a sandwich is two slices of bread with meat in between."* Although this definition may be a good candidate, it misses important features that make a sandwich a sandwich. In this analogy, our proposed approach would ask the user if grilled cheese is a sandwich as a counterfactual proposition. This counterfactual highlights the discrepancy between the outcome of executing the proposed rule (grilled cheese *would not* be a sandwich because of the lack of meat) and a pre-trained model's understanding (grilled cheese IS a sandwich according to the LLM). By highlighting the difference between the original definition (having meat in between) and a new synthesized definition with the new counterfactual example annotated (having any filling in between), the two entities are able to iteratively negotiate and reach a shared definition of sandwiches. This iterative redefinition would continue along different feature dimensions of a sandwich (like the number of breads, the type of bread, etc.) until an optimal shared agreement is reached. It is worth noting that this is a collaborative negotiation rather than a debate. Neither of the parties came to the interaction with a

complete definition of sandwiches and tried to convince the other party. Instead, they came with an incomplete definition, were open to reflect on and change their own conceptual understanding based on interactions, and shared the goal of reaching clearer boundaries of the target concept.

**Structure Aligned Data Rendering.** Mocha employs Structural Alignment Theory (SAT) [27] to support the user's cognitive process of interpreting and understanding varying generated data. According to SAT, humans compare two similar entities by trying to find structural alignments between them, and then comparing corresponding elements, with a special focus on differing aligned elements. Mocha contrasts the original user-labeled data with generated counterfactual examples, by visually emphasizing portions of the counterfactuals that have changed over the parts that stayed the same. Specifically, Mocha displays unchanged elements of the counterfactual examples in gray. In contrast, elements that have changed—and may thus influence a change in label—are highlighted in black, drawing the user's attention to these critical differences. By assisting users in comparing discrepancies between their own label definitions and the neuro-symbolic model's learned decision boundaries, users can provide annotated data that can update the model to align with their expectations.

We validated the usability and effectiveness of Mocha in a lab study with 18 participants. Participants reported that the tool's workflow enhanced their understanding of both the underlying model's behavior and the data itself. Mocha was also shown to significantly improve annotation efficiency and improve the model's performance in learning user intents. These findings point to important design implications for future human-AI alignment efforts. Specifically, they underscore the need for co-adaptive systems that can evolve along with users' mental models and definitions of labels. Our findings also highlight the implications of closing the loop in supporting human cognition with proposed interactive ML pipelines. This adaptability is crucial for fostering deeper and more effective interactions between humans and artificial intelligence in complex data environments.

This paper makes the following contributions:

(1) We contribute a SAT-based rendering method for counterfactual examples.
(2) We built Mocha to understand how Variation Theory-based counterfactual generation [24] combined with SAT-based counterfactual rendering affects the human's experience in co-adaptive machine teaching.
(3) A lab study with 18 participants to demonstrate the usability of Mocha and its effectiveness in improving annotation efficiency, enhance the model's learning, and facilitating co-adaptive learning where users gain insight into the states of the model and reflect on their own understanding.

## 2 Related Work

### 2.1 Human-AI Alignment through Data Annotation

Alignment is a bilateral process; it refers not only to AI acting according to human intentions but also to humans better leveraging AI by understanding the mechanisms behind it [54]. In this process, both the trainer and the learner aim to develop and maintain an accurate understanding about the target concept. In the machine learning pipeline, one way users show their intentions is through annotating data [66]. However, when labeling under uncertainty or in the initial training phases, users may lack an understanding of the model capabilities.

Machine teaching, a part of the human-in-the-loop approach, has been used as a process in which a human expert (the "teacher") provides guidance to a machine learning model to help it learn important and robust features for decision making [57]. In this workflow, humans continuously guide the model to align its learning with their intentions. A common way to do this has been through data annotation [68]. To align model training with human intent through data annotation (1) the human needs to understand the current state of the model and (2) the human should be able to take action to steer the model in their desired direction [66]. This signifies that alignment demands not only that humans should be able to steer AI in their desired direction but also that humans need to understand the current state of AI and what it has learned in order to better utilize the latest AI advancements.

Previous work shows that, although incomplete, users may possess some knowledge about target concepts [58, 59], which they use as a reference when building classifiers IML [10, 56]. AnchorViz [56] and Alloy [10] provide more context to the users through clustered data to help them determine what should and should not belong to different labels. As users interact with more context and data, their definition of the concept could shift and evolve [40]. A key challenge here lies in designing interactive systems that both acknowledge the dynamic nature of users' conceptualizations and transparently illustrate how these evolving inputs influence the model's outputs [50]. Consequently, this also requires the model to demonstrate how it adapts to the user's preferences to enhance its interpretability and increase human trust in the system [36].

### 2.2 Active Learning and Counterfactuals

Active Learning (AL) in machine learning is an approach in which the learning algorithm selectively chooses which data points should be labeled for training [20]. The primary goal of this approach is to minimize the amount of labeled data needed to learn a target concept by requesting labels, usually from humans, for the most informative examples, allowing the concept to be learned with fewer annotations [3]. To accomplish this, pool-based [69] and instance synthesis-based (also called query synthesis) [52] selection strategies have been used. An example synthesis-based strategy can be especially effective in domains with ambiguous and subjective labeling, as it creates new, potentially informative examples that broaden the distribution of labeled data while continuously adapting to the model. Counterfactual generation can be seen as a more targeted form of query synthesis. While query synthesis broadly creates new examples to inform the model, counterfactuals specifically explore "what-if" scenarios by modifying existing instances in meaningful ways [51]. Counterfactuals have been used to test the sensitivity of a model to small changes to refine its understanding of causal relationships [19]. Both query synthesis and counterfactuals aim to generate examples that are useful to the model. However, not all examples are equally informative for the model or equally easy to label for humans [13].

Due to the inherent complexity of language and its discrete nature, natural language counterfactual generation presents greater difficulties compared to structured and image data. For that reason, natural language counterfactuals have seen limited exploration. Previous works have proposed both generation-based and augmentation-based approaches. Schumann and Rehbein [52] uses variational auto-encoders to generate examples from uncertain regions in a model's latent space to improve a classified model. Alternatively, Dixit et al. [14] uses a retrieve-then-edit framework to generate counterfactuals by conditioning on naturally occurring data. With the generative capabilities of large language models (LLMs), there have been more efforts to automatically generate plausible counterfactuals by augmenting real examples [11, 14, 63]. Polyjuice [63] uses a fill-in-the-blank approach to generate counterfactuals by perturbing specific parts of a sentence according to predefined control codes (e.g., negation, quantifiers, or lexical modifications). Similarly, DISCO [11] uses spans to determine what needs to change. Although both approaches use different methods to determine what needs to change, they find that the counterfactuals improve the downstream model's performance.

Most previous research in counterfactual generation has focused on the model side by either generating counterfactuals to improve the model's performance or explaining its behaviors post hoc. As intended with the design of Mocha, we believe this process should encourage users to engage in analogical thinking, enhancing their reflection and understanding of the underlying concepts when they are not well defined. Mocha uses human cognitive learning theories to support human annotation efforts for model training. Specifically, we use Variation Theory of learning [44] which states that for learning to occur, some aspects that define the concept being learned must vary while others are held constant.

## 2.3 Supporting Sensemaking Based on Variation Theory and/or Structural Alignment Theory

Structural Alignment Theory (SAT) [27] is a cognitive theory that explains how people make sense of concepts by comparing relational structures between two items. It states that understanding and sensemaking involve mapping the relationships between elements, especially in complex and ambiguous tasks. While SAT focuses on similarities and differences within alignable structures, Estes and Hasson [17] highlights the significant role of bringing salience to "non-alignable" differences. In decision making, SAT argues that people tend to focus on alignable differences—features that can be directly compared—rather than on differences that cannot be easily aligned. Although its application remains limited outside of the field of psychology, SAT has been used in broad domains such as consumer behavior research [37], spatial data analysis [12], Human-Robot Interaction (HRI) [6], and Human-Computer Interaction (HCI) [64]. The last two prior works also combine Variation Theory (VT) and SAT together, as we did (i.e., a corollary of SAT referred to as Analogical Transfer/Learning Theory).

In developing ML models, annotators often engage in a process of comparing instances within the data, not just to match surface-level features such as keywords, but to discover relational patterns that inform their label definitions and boundaries [41]. Therefore, SAT provides a useful and applicable framework for thinking about data annotation, particularly in domains where annotators continuously define and refine labels during model training.

Comparison as a means for sensemaking also finds relevance in modern tools designed for large-scale output sensemaking; two previous tools explicitly leverage SAT and Variation Theory (VT)-based designs. Positional Diction Clustering (PDC) [29] is a structure mapping engine [28] introduced to facilitate sensemaking of many LLM responses to the same or similar prompts. It finds a structural mapping across all the LLM responses and can highlight alignable differences within that alignment using text salience. ChainForge [2] provides an interface to compare model outputs, where the variables that create the systematic variations in models and model prompts correspond to dimensions of variation in Variation Theory. Both systems enabled users to quickly identify variations and patterns within the data and support exploration and hypothesis testing.

In line with previous work, Mocha aims to support a user's efforts in the disambiguation of concepts through structural comparisons of counterfactual data in the context of machine teaching. Specifically, Mocha highlights variations between data items to help users identify inconsistencies between their own label interpretations and the model predictions. In the context of interactive ML, where users are in charge of labeling data with the goal of influencing the model's training, interactive error correction is crucial [9, 55]. By presenting relational structures (e.g., causal chains for wrong predictions in counterfactuals) instead of just showing learned feature importance, Mocha helps users understand how the system makes decisions and identify how their annotations could change the model's behavior.

# 3 System Description

Machine teaching with exploratory data labeling requires users to provide distinguishing training examples of a concept [33]. When labeling similar data with subtle differences, users must compare data points while incorporating concept definitions [62]. In the context of co-adaptive learning, supporting the intertwined evolution of both the user's understanding and the model's learning is crucial [16].

To support users in providing informative training examples based on the model's current state, the back-end pipeline of Mocha integrates a neuro-symbolic approach with LLMs to guide the synthesis of counterfactuals that resonate with human cognitive processes. Building on methods proposed in PaTAT [24], Mocha first generates human-readable neuro-symbolic pattern rules from partially labeled text data for classification. Then, Mocha generates synthetic counterfactual text data that share syntactic and semantic patterns with the original text data, yet differ in the outcomes of the labels predicted by an LLM. This approach allows users to annotate data near conceptual boundaries, improving their understanding of the current limitations and capabilities of the neuro-symbolic model, as well as possibly facilitating the evolution and refinement of their own notions of the concepts involved. These annotations are pivotal for the model to learn user-specific values, preferences, and goals.

Counterfactual generation based on Variation Theory allows the implementation of Structural Alignment Theory on Mocha's interface design to render and highlight alignable differences between original and generated data items in real time. This facilitates the user's sense-making process of discerning key differences among items and reasoning about how and why they lead to different labels during annotation. Thus, ultimately improving the efficiency and effectiveness of the model teaching process.

The section begins by outlining the design goals of Mocha, presents a motivating user scenario, discusses its key features, their rationale, and how they connect to the design goals, and concludes with implementation details.

## 3.1 Design Goals

Mocha is designed to assist the user in their data annotation efforts and sensemaking while simultaneously providing useful training data for the model's learning. Specifically, Mocha follows these design goals:

***DG1: Facilitating User Understanding of the Model's Current State.*** Teaching is inherently exploratory. Aligning model training through MT with human values iteratively requires clear user understanding of the model's current state [55]. This is achieved by providing transparency in the model's decision-making processes, emphasizing areas of uncertainty or salience, and illustrating the model's data interpretations. Such insights enable users to recognize where the model excels, where it falters, and how it evolves over time, empowering them to make informed decisions to refine and enhance the model through targeted annotations.

***DG2: Augmented Data Should Refine the Model's Decision Boundary .*** The impact of labeled data is crucial in refining the model's decision boundary, particularly within areas of high uncertainty [65]. Therefore, the data annotated in each iteration should improve the model's understanding of its decision boundaries. Augmented data should spotlight edge cases and ambiguous instances near the decision boundary [8]. Addressing these cases improves the model's generalization capabilities and offers users deeper insights into its decision-making processes.

***DG3: Enhancing Interface Support for User Annotation of Generated Data.*** Research has identified data annotation as a critical bottleneck in the model training pipeline [4]. While the original data is a fixed resource, augmenting data can be strategically aligned with users' cognitive processes to facilitate sensemaking. Traditional annotation methods, which rely heavily on manual reviews, often fall short in handling complex datasets and counterfactual examples. To address this, Mocha's initial phases of data integration employ interaction mechanisms that simplify the identification and annotation of data points. These mechanisms contribute to a more diverse and comprehensive dataset and clarify the intricacies of decision boundaries. This requires the development of interfaces and visualizations that demystify the generated data, allowing systematic variation and coverage across the concept space.

## 3.2 Motivating User Scenario

Alice is working on training a model to classify text snippets from online reviews. Although she has an idea of the possible labels in the dataset, there is some uncertainty about which features are most relevant for the model's training and how these features define the feature space for each label. She is also exploring how to differentiate between seemingly overlapping concepts, for example weather friendliness of a restaurant staff should belong to the label *service* or *environment*. Alice decides to use Mocha to iteratively label data and train a classifier model. While she iteratively trains the model she expects to have a more complete understanding of what appropriate features each label should contain and have a solid definition of the labels.

Alice uploads her dataset into Mocha, a tool designed to iteratively align the model's learning process with the user's mental model. She begins the process of assigning labels to data items. After every ten annotations, Alice notices the tool suggests labels to unlabeled data items based on learned pattern rules (Fig 2-A). These rules, generated by a neuro-symbolic model, are constructed using program synthesis to find an optimal combination of domain-specific language that best fit Alice's labeled positive and negative examples (details in Appendix A.3). The rules aim to capture the discerning features of each label based on Alice's annotations. While some of the suggested rules align with Alice's mental model, others may be too broad or fail to accurately reflect her intended label definitions.

To see data with suggested labels (Fig 2-C), Alice can click on a pattern rule (Fig 2-A) to filter data points that match the selected pattern rule (*DG1*). As Alice clicks on the data point to assign a label to it, Mocha generates counterfactual examples (Fig 2-E) that are structurally close to the original data point she is currently labeling. The generated counterfactual examples match the learned neuro-symbolic rules (Fig 3-D) but are labeled differently by an off-the-shelf LLM (Fig 3-C, *DG2*).

After labeling the original data point (Fig 2-B), Alice moves on to label the generated counterfactual examples. When examining them in batch, her attention is drawn to the differences between the original and counterfactual examples, with the differing parts highlighted in **black** and the unchanged parts in *gray* (Fig 3-E and F, *DG3*). Each of the generated counterfactuals match the learned patterns and are incorrectly labeled by the neuro-symbolic model. With the generated counterfactual examples, Alice sees what the model is learning and failing to learn. By labeling the counterfactual examples, Alice provides the model with additional data points to fortify the interpretation of a label, as shown by the learned pattern rules. At the same time, making labeling decisions on data points that are at the model's decision boundary helps Alice refine or confirm her own understanding of the label. The labeled counterfactual examples are then used during consecutive model training. After each round of annotation the neuro-symbolic model updates its learned rules and this process continues until Alice's interpretation of each label finally aligns with the model's.

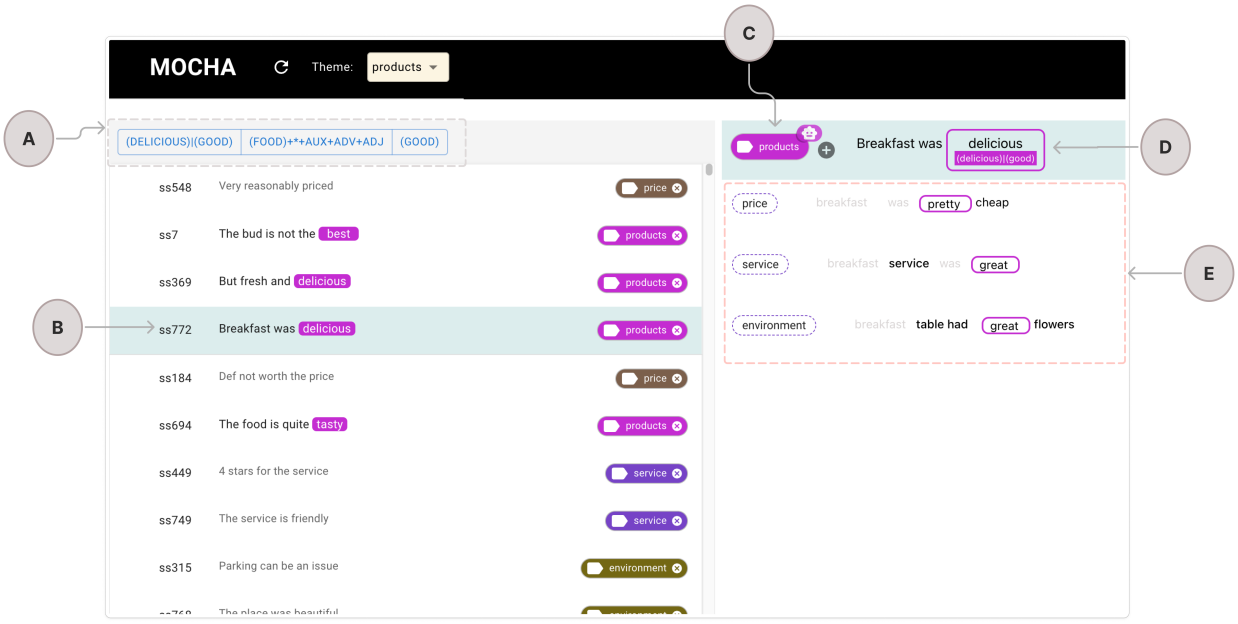


**Figure 2: Mocha uses neuro-symbolic pattern rules (A) to generate counterfactuals. For each example labeled by the rules (B), Mocha generates counterfactual examples that match the original patterns (D) but belong to a other than the original label (C). The generated counterfactuals are then rendered below the original example with highlighting of what has changed and what has stayed the same (E) for each alternative label.**

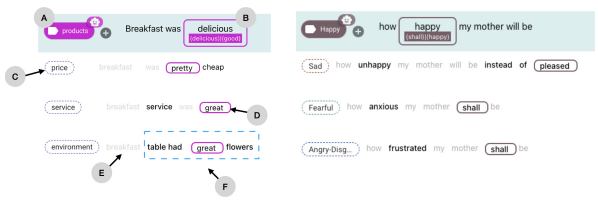


**Figure 3: Mocha facilitates analogical reasoning using visual cues. For each model-labeled example (A) and its corresponding learned neuro-symbolic rule (B), counterfactual examples are generated for a set of target labels (C). Phrases consistent with the original example are displayed in gray text (E), while varying phrases are displayed in black text for visual salience (F). Additionally, the text of the counterfactual that would mislead the neuro-symbolic model into classifying it as the original label (by matching the original label's rule) are highlighted in the theme color (D), helping users understand how their annotations contribute to model updates.**

## 3.3 Key Design Features

This section describes the following key design features of Mocha to support the continuous training and alignment between humans and AI.

*3.3.1 The Generation of Alignable Counterfactuals.* Counterfactual examples enhance the training of a model by generating misclassified instances, which the model can then correct through retraining. This process works because counterfactuals reveal edge cases in the model's learned decision boundaries. Gebreegziabher et al. [24] argued that counterfactual generation that follows the principles of VT allowed the introduction of discriminatory variance for the model to learn on. According to Variation Theory, learners better understand concepts by observing variations along critical features (dimensions of variation) that define that concept and, separately, observing variations along superficial features that do not define that concept—all while other features, when possible, are held constant. For instance, students who see triangles in different orientations can deduce that the defining characteristic of a triangle is its three sides and not its orientation. Building on that, in Mocha, we adopt this approach to develop a user-facing interface that supports a user's learning in parallel to the model's learning.

To integrate VT into counterfactual generation, this method begins by identifying key features and introducing variations while still satisfying the predefined neuro-symbolic pattern rules [25]

that currently define the machine-learned concept; these pattern rules are interpretable features the model has learned as critical for a given concept (see details in Appendix A.3). The rules encompass lexical, syntactic, and semantic elements—including part-of-speech tags, word stems, synonyms (soft matches), and entity types—organized in regex-like patterns. These patterns help capture commonalities across datasets with similar labels (Fig 2-A).

In other words, for each label, the learned neuro-symbolic patterns reflect the model's current interpretation of that label; data points that match the model's linear combination of the neuro-symbolic patterns would be classified with that label. Therefore, when the learned patterns are inaccurate, the generated counterfactuals should provide examples that match the sub-optimal pattern but likely correspond to a different label. To facilitate generating counterfactuals along this dimension of variation, our approach starts by prompting an LLM to generate candidate phrases that match the learned pattern (Appendix A.4). For example, for a data item that was labeled "product", the sentence *"Breakfast was **delicious**"* that matches the pattern *'(delicious)|(good)'* will have phrases *['well priced', 'pretty cheap', 'worst deal', 'good but overpriced']* generated as candidate phrases for a target label "price". The candidate phrases are used to enforce that the augmented counterfactual always includes the learned neuro-symbolic pattern. In the counterfactual generation prompt (Appendix A.5), we explicitly instruct the LLM to modify the original example, making minimal changes while still including the generated candidate phrases, to change the original label into a set of different target labels (see Algorithm 1) (DG2).

**Algorithm 1** Generates counterfactual data based on learned neuro-symbolic patterns

**Require:** Original dataset $D$, User label $UL$ ,Target label $L$

**function** GenerateCounterfactuals($D, UL, L$)
  Initialize $D_{cf}$ as an empty dataset
  **for** each $d \in D$ **do**
    $p_d \leftarrow$ GetSymbolicPattern($d, UL$) ▷ Generate patterns based on the user assigned labels
    $candidatesPhrases \leftarrow$ GenerateCandidatePhrases($d, p_d, L$) ▷ Generate phrases that match the pattern but are about target label $L$
    $variations \leftarrow$ GenerateVariations($d, candidatesPhrases, L$) ▷ Change parts of the sentence with one of the candidate phrases
    **for** each $v \in variations$ **do**
      $cf \leftarrow$ CheckPattern($v, P$) ▷ Check if counterfactual matches pattern $P$
      **if** $cf$ successfully flips the label to $L$ **then**
        Add $cf$ to $D_{cf}$
      **end if**
    **end for**
  **end for**
  **return** $D_{cf}$
**end function**

*3.3.2 Facilitating User's Perception of Similarity and Differences.* Assisting users to perceive consistencies and variations across different data points enables them to quickly develop an accurate mental model [29, 48]. To support this process, Mocha draws design inspiration from the Structural Alignment Theory [27] of how humans compare and contrast objects. Structural Alignment Theory states that humans naturally look for structural mapping between representations of objects to help them understand, compare, and infer relationships between said objects. In our context, these objects are the original examples and their counterfactuals. In rendering the generated counterfactuals, Mocha facilitates analogical reasoning through the mapping of the counterfactual label (Fig 3-C) and the generated counterfactual.

To assist users in assessing the appropriateness of the counterfactual label for the generated example, Mocha uses visual cues enabled by the structure of variation induced by the Variation Theory-based counterfactual generation method in the previous step. Specifically, the changes introduced to change the original example into the counterfactual are highlighted in black (Fig 3-F) to draw user attention to them. This black text stands out more prominently than the unchanged text, which is rendered in gray (Fig 3-E).

To determine this mapping, Mocha adopts the Levenshtein distance algorithm [67], which calculates the minimum number of edit operations at the word level required to transform the original example into the counterfactual example generated. Specifically, we define two types of edit operations: additions (inserting words) and deletions (removing words). The algorithm splits each sentence into its component words and identifies the shortest sequence of operations to transition from one sentence to another. Our objective is to minimize the number of operations while striving to maintain the longest continuous phrase unchanged between the two sentences. For example, given the original sentence *"Breakfast was delicious"*, and a counterfactual sentence *"Breakfast was pretty cheap"*, the algorithm would identify a delete operation for the word 'delicious' followed by an insert operation for 'pretty cheap'.

*3.3.3 Comparison Through Carried Over Matched Neuro-symbolic Rule.* Mocha aims to facilitate the user's understanding of where and why the neuro-symbolic model's understanding diverges from their expectations. To support this, it leverages the executable nature of the learned neuro-symbolic rules. Mocha highlights phrases in generated counterfactual that match the learned 'imperfect' neuro-symbolic (Fig 3-D). This process can be understood as the common relational structure between the original and counterfactual examples.

A key visual aid in this process is the use of theme colors (Fig 3-D), which highlight parts of the counterfactual that could have misled the model into making incorrect classifications. By applying a consistent and striking color to these terms, the system visually projects the model reasoning process onto the interface, making the inference projection process possibly easier to understand for users. From a cognitive perspective, the theme color aligns with the human's (theorized) structural mapping engine [27] by making relational discrepancies between the original and counterfactual examples more explicit. The model's reasoning is "projected" onto the counterfactual, enabling users to easily see which aspects of the counterfactual match the model's existing rules, and which aspects

lead to erroneous inferences (DG1). This immediate feedback supports users in correcting the behavior of the model by adjusting the labels and refining the classification boundaries through targeted interaction.

## 3.4 Implementation

The interactive Web application of Mocha was developed using React[1]. The backend server utilized Python's FastAPI[2] framework to facilitate communication with OpenAI's API and to the frontend. In the backend, candidate phrases and counterfactuals were generated using the GPT-4o model from OpenAI through API calls. We used Firebase[3] to track and store participant's interaction log data. Both the front-end and the back-end were hosted on a Google Cloud[4] server for the user study.

# 4 User Study

The user study[5] aims to evaluate the effectiveness of Mocha's key features, informed by Variation Theory (VT) and Structural Alignment, on augmented data annotation and bi-directional alignment. Specifically, we evaluate the efficacy of the augmented counterfactuals on the user's annotation efforts, the model's learning, and the user's learning about the data and the relevant concepts.

The study specifically investigates the following research questions:

- **RQ1:** Can Structural Alignment-based counterfactual rendering improve efficiency and lower cognitive load during the annotation process?
- **RQ2:** How useful are Variation Theory-based counterfactual generation in allowing the model to learn about the user's intents?
- **RQ3:** How useful are Variation Theory-based counterfactual generation and Structural Alignment Theory-based counterfactual rendering in allowing the users to learn about the data and clarify their intents to the model?

## 4.1 Participants

We recruited 18 participants (11 male, 7 female) with varying levels of experience in developing and training ML models, as detailed in Table 2 in Appendix A.1. The participants' ages ranged from 18 to 34 years. Among them, 4 participants self-reported having a beginner level with ML, 10 were intermediate, and 3 were experts. One participant had no previous experience with ML. More information on the demographics and backgrounds of the participants can be found in the Appendix A.1.

## 4.2 Study Procedure

Each study session lasted approximately 90 minutes and was conducted either in-person in a usability lab or virtually via Zoom (3 in-person, 15 virtual). After obtaining informed consent, the participants received a 5-minute tutorial on the key features of the tool. Participants were asked to train a multi-class classifier model by assigning one or more labels to data-items from the given list of labels. While participants worked with the same set of labels, they were told to follow their own interpretations of both the labels and data points. The study used a think-aloud method, asking participants to verbally express their thoughts while annotating the data. The participants were then assigned one of two datasets and engaged with Mocha in three 25-minute sessions under different conditions. Here, they observed the neuro-symbolic model retrain and update following each annotation round. The sequence of these conditions was randomized, and details about the order and specific datasets are available in Table 3 in Appendix A.2. After each condition, participants completed a NASA-TLX [34] survey to assess their cognitive load. The study ended with a post-study questionnaire on Mocha's perceived usefulness and usability, and a semi-structured interview exploring the use of tools and experiences of participants in different conditions.

The study aimed to observe the participants' abilities to refine their subjective definitions of labels and the model's effectiveness in learning from the labeled counterfactual data. Initially, we presented participants with a dataset that included a predefined set of labels, allowing them the flexibility to define and redefine these labels as they annotated examples.

*4.2.1 Datasets.* In this study, we selected two datasets that are open to subjective interpretation and do not necessitate domain-specific knowledge from participants. Each participant worked with only one dataset throughout all conditions. The sampling method for each condition was designed so that no participant would encounter the same data item more than once across different conditions.

- **Emotions** [1] - Each entry in this dataset consists of a text segment extracted from tales with labels that indicate the predominant emotion conveyed. Participants worked with 5 categories—fearful, sad, happy, surprised, and anger-disgust. This dataset contained 100 independent samples for annotating and testing for each condition.
- **YELP** [5] - This dataset consists of user reviews of businesses (e.g., restaurants, retail stores) collected from Yelp with 4 categories—price, service, environment, and products. This dataset contained 160 independent samples for annotating and testing for each condition.

*4.2.2 Conditions.* The study used a within-subjects design, where each participant experienced the system under three different conditions. To minimize carryover effects, the order of the conditions was varied for each participant.

- **Condition 1 (Non VT Counterfactuals)**: Participants were asked to label counter examples generated without the use of VT and neuro-symbolic patterns. The generated counter examples were displayed without highlighting alignable differences for users to label.
- **Condition 2 (VT Counterfactuals without alignment)**: Participants were asked to label counter examples generated using VT with neuro-symbolic patterns but without highlighting alignable differences.
- **Condition 3 (VT Counterfactuals with alignment)**: Participants were asked to label counter examples generated

[1]https://reactjs.org/
[2]https://fastapi.tiangolo.com/
[3]https://firebase.google.com/
[4]https://cloud.google.com/
[5]The study protocol was reviewed and approved by the IRB at the lead author's institution, where the study was conducted.

using the same pipeline as *Condition 2* with alignable differences highlighted in the interface.

## 4.3 Study Results

*4.3.1 Data Analysis.* We conducted statistical tests to compare responses to all the Likert scale survey questions, as well as the time spent annotating counterfactuals for each condition.

To analyze the annotation efficiency, we first conducted a Kruskal-Wallis rank sum test [39] to determine if there were statistically significant differences in annotation time across the three conditions, because our data violated the homogeneity of variances assumption, making non-parametric methods more appropriate. To compare each condition against each other, we conducted a post-hoc pairwise test using the Wilcoxon rank-sum test [47] with continuity correction and Bonferroni adjustment.

To analyze the Likert scale ratings of the participants from the post-study questionnaire, we first performed a Friedman test [23] to determine whether there were statistically significant differences in participant ratings across the three conditions. Following this, we used Wilcoxon signed-rank tests with Bonferroni correction [47] and Kendall's W [21] for post-hoc pairwise comparisons. These tests are nonparametric, which is appropriate for the ordinal nature of Likert scale ratings.

The analysis of the interview results was done through open coding [61], in which two members of the team coded the interview transcripts independently and then came together to consolidate.

*4.3.2* ***RQ1:*** *Impacts of Structural Alignment interfaces on annotation efficiency and cognitive load.*

*Improved annotation efficiency.* To understand the impact of alignable differences on the participant's annotation efficiency with the generated text, we compare the average time it took for participants to read and make their first annotation on a generated sentence against the time it took them to complete a batch of annotations (i.e., annotating all generated examples associated with an original sentence as seen in Fig 3). We calculate time of annotation as the difference between when the generated counterfactuals are rendered on the screen to when the participant assigned the first label, and subsequently all consecutive labels in that view.

On average participants annotated more data points in condition C3 (84.6, SD=26.0) compared to both C1 (70.82, SD=31.2) and C2 (69.65, SD=31.0). As shown in Figure 4, participants spent a similar amount of time to annotate the first data point they saw across all three conditions. However, the batch annotation times were notably shorter in Mocha's counterfactuals (C3). The batch annotation times for counterfactuals with alignable differences (C3) were significantly lower compared to both counterfactuals generated without Variation Theory (C1) ($p<.0001$) and those generated with Variation Theory but without highlighted alignable differences (C2) ($p<.0001$). There were no statistically significant differences between C1 and C2 in their batch annotation times. This suggests that the efficiency improvements are primarily attributable to the specific features of C3, which is the rendering of alignable differences between the generated counterfactuals and the original example. Despite the lack of statistical difference between C1 and C2, the application of Variation Theory in C2 played a crucial role, enabling the introduction of alignable differences in C3.

*Participants Perceived Mocha as More Useful in Condition 3.* In the post-condition questionnaire, participants compared their experiences across three different conditions. Figure 5 shows their subjective assessments in four dimensions: knowledge gained about the data, the degree of control they felt in changing the behavior of the model, the usefulness of the system, and the ease of using the system. C3 emerged as notably the superior condition, with participants reporting the greatest gains in knowledge about the data, increased control, and enhanced usefulness of the system compared to the other conditions.

Contrary to the reduced batch annotation time, we did not find a statistically significant difference in participant ratings of their perceived ease of use across the three conditions.

However, we find statistically significant advantage of C3 in the other three measures ($p<.05$). The participant's rating of the knowledge gained in the three conditions revealed a statistically significant difference ($X^2(2, N=18)=11.6$, $p=.002$) compared to the other conditions. The effect size, measured by Kendall's W, was 0.323, suggesting a moderate level of agreement among the rankings. The post-hoc pairwise test showed that the participants rated their level of knowledge gained about the data in C3 significantly different from both C1 ($p=.02$) and C2 ($p=.007$), while there was no significant difference between C1 and C2. This indicates that C3's structurally aligned rendering led participants to feel they gained significantly higher level of knowledge compared to the other conditions. These results align with the participant's usefulness ratings for the three conditions ($p=.0009$) with an effect size of 0.39 suggesting a moderate to strong level of agreement among the ranking. Similar to the knowledge gained ratings, we find no significant differences between C1 and C2 but significant difference between C3 and both C1 ($p=.01$) and C2 ($p=.002$).

*No difference in cognitive load observed.* The NASA-TLX scores revealed no statistically significant differences in cognitive load between conditions (see Figure 7 in the Appendix A.6). This suggests that the variations in the counterfactuals generated and their presentation in C3 did not impose additional mental demands on the participants compared to C1. More research is needed to determine whether the interventions introduced in C3 could reduce cognitive load over extended use.

*4.3.3* ***RQ2:*** *Effect of Variation Theory based counterfactuals on the model's learning.* A major goal of data augmentation is to improve the generalizability of the model [14]. To that end, the annotated data should contribute a meaningful distribution to the model's training dataset to align with the user's own intentions. To assess the downstream impact of the generated data on model training, we evaluated the model's performance when trained only on the real data and compared it to the model's performance when trained on the combination of the real data and the labeled counterfactuals. The participants' dataset, including the counterfactuals, provided a basis for comparing model performance under two conditions: with and without counterfactuals. Given that participants could flexibly define their own interpretations of the data, we use each

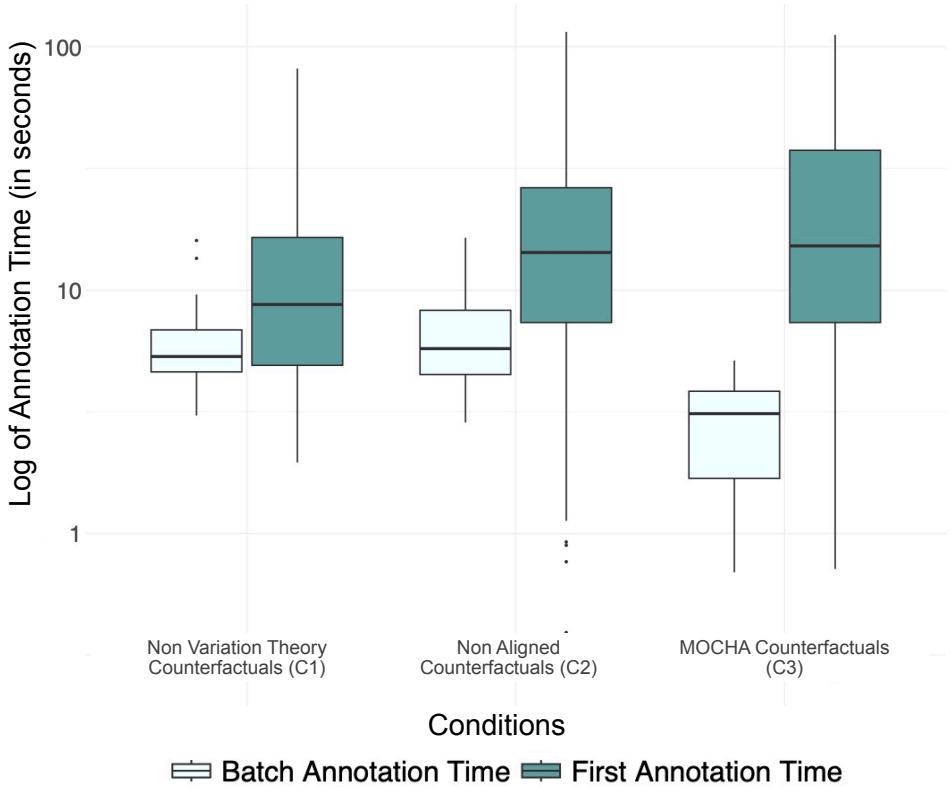


**Figure 4: Comparison of average annotation times under the three conditions. The table shows that while VT-based counterfactuals increase the time for the first annotation, SAT-based rendering significantly reduces the time for annotating each data point in the batch.**

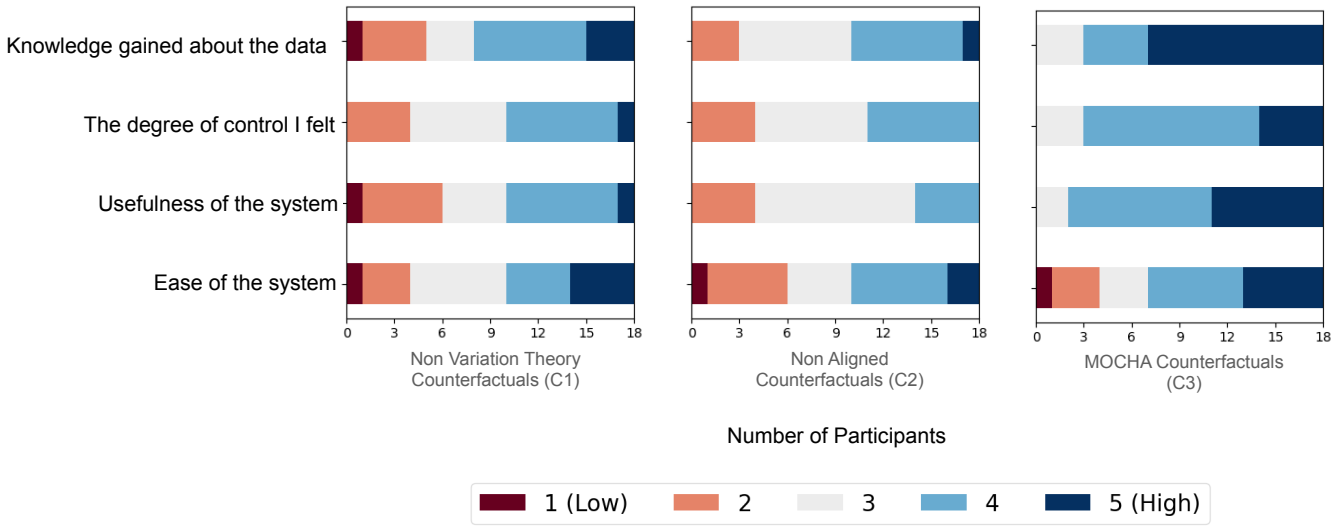


**Figure 5: Participants' response to post study questionnaire comparing the three conditions.**

participant's final labels as the ground truth. We calculated the precision and recall of the model using the finalized labeled data.

In our experiment, we observed that the participants had moderate agreement with each other, measured by Fleiss' Kappa [22] (Yelp=0.67, Emotions=0.41). This relatively low agreement score for

the emotions dataset suggests that the participants did not reach a strong agreement among themselves about how to label the data indicating inherent ambiguity and subjectivity in the task. However, when the model is trained on these somewhat inconsistent labels and evaluated for agreement with their corresponding participant individually (using Cohen's Kappa [45]), the averaged agreement between the participant and their final model is stronger (Yelp=0.76, Emotions=0.87).

Table 1 presents a comparison of the final model performance for each participant. In most cases, the inclusion of labeled counterfactuals contributed to increases in both precision and recall. The inclusion of counterfactuals often resulted in a substantial increase in precision, indicating that the models were better able to correctly classify relevant instances while reducing false positives. This improvement suggests that the counterfactuals provided essential information that helped refine the models' decision boundaries.

In scenarios where real data lacked annotated labels, the labeled counterfactual examples were instrumental in initiating the model's learning process and enabling the generation of relevant neuro-symbolic rules. For example, P4 observed the model learned pattern rules for the label *'service'* from labeled counterfactuals generated for the label *'price'*. Initially, there was insufficient annotated data for the model to generate patterns for the label *'service,'* but with the inclusion of annotated counterfactuals, the model successfully learned a neuro-symbolic rule for the label *'service'* which was *(friendly)+*+NOUN* [6]. Similarly, P16's process of labeling and retraining the model revealed a transformation in both their mental model and the model's learned patterns. Initially, P16 viewed the model's learned patterns for the label *'sad'* as "generic", relying on rules such as '*$PERSON*' (matching all entities under person) and '*PROPN|(mourn)*' (matching all sentences with proper nouns or synonyms of mourn). As the model was retrained with more targeted examples, P16 observed the emergence of more specific patterns, such as the rule *(weep)++NOUN*, which matched phrases strictly involving synonyms of "weep" followed by a noun.

While this was a common change in how the generated counterfactuals allowed the model to learn pattern rules with higher precision, some participants observed the model was giving correct labeling decisions but learning wrong patterns (i.e., write for the wrong reasons). For instance, P10 noticed that in the early stages of training, the model learned the pattern *(sister)*[7] for the label *'fearful.'* Recognizing this as overfitting to the available labeled examples, P10 adjusted their strategy to focus on labeling counterfactuals. Specifically, they labeled examples where the pattern *(sister)* occurred but were labeled with concepts other than fearful. This iterative approach eventually led the model to learn more appropriate patterns, such as *(frighten)|(stand)* and *(small)*, which aligned better with their intended concept of *'fearful'*.

The improvement in recall, although present, was more inconsistent across participants. In some cases, the integration of counterfactuals significantly enhanced the model's ability to identify relevant instances that were overlooked in the original dataset. For instance, P18's final model was able to learn the rule *[sense] | (frightened)*[8] When trained with counterfactuals in addition to the original pattern rule *(little) | (dread)*, it had learned with just the original examples for the label *'fearful'* ultimately increasing the recall. Similarly, for the label *'environment'* we observe the model's generated pattern rule change from *(great)+(place)* into a combination of *(atmosphere)|(area)* and *(cozy)|[dining]* for P6. These cases illustrate how counterfactuals can enhance the flexibility and precision of pattern learning, leading to improvements in recall. However, the degree of improvement varied across different datasets and participants. Notably, the model performance for P18 actually decreased in recall, suggesting that the counterfactuals provided may have been less impactful, possibly because P18 had already achieved high performance without them. Further research is needed to explore these dynamics and understand the variable impacts of counterfactuals on model performance.

Overall, the incorporation of counterfactuals has generally improved the models' F1 scores, driven largely by the improvements in precision. This suggests that counterfactuals have effectively improved performance without necessitating a significant trade-off between precision and recall. However, the less consistent increase in recall underscores the necessity for further research into how counterfactuals can be designed or selected more effectively to uniformly boost both aspects of model performance.

#### 4.3.4 ***RQ3:*** *Co-adaptive learning with VT-based counterfactuals generation and Structural Alignment Theory-based counterfactual rendering.*

*Effects of alignable mapping in participant reading behavior.* Through the think-aloud sessions, we observe a noticeable shift in reading behavior when participants were annotating data points under C3, which came with highlighted alignable differences. In this condition, parts of the text were visually grayed out to denote redundant content from the original example, leading participants to generally skip these sections in their think-alouds. Instead, they concentrated on and vocalized sections of the text that remained bolded, representing new or added content in the generated counterfactual. This pattern of selective attention suggests that the visual cues provided by Mocha effectively guided participants to focus on more relevant information within the context of unchanged text when making their labeling decisions. It is worth noting that P3 and P8 mentioned feeling more comfortable with the more familiar visualization in C1 and C2 during their first impression of the conditions. When comparing their initial impression on the conditions, P8 said: *"as I get familiar with this system [C3], I feel more skilled"* to use the highlighted and grayed phrases.

*Counterfactuals that followed Variation Theory enhanced model validation and retraining.* Participants mentioned that counterfactuals adhering to Variation Theory significantly aided their understanding of the model's current learning state. Some participants used counterfactuals to validate the model's understanding of specific labels. For example, P3 described their approach: *"I think that is how I am using the counterfactuals to verify that it has actually learned what is the key part that makes this a product? And so if it just gives me [counterfactuals] [that are] probably still product,*

[6]This rule matches any sentence that has synonyms of the word friendly followed by a single wildcard and a noun

[7]This pattern rule matches sentences that include any synonyms of the word *sister*

[8]This pattern rule matches all sentences with the literal word *'sense'* or any synonyms of the word *'frightened'*.

| PID | Without Counterexamples | | | With Counterexamples | | |
|---|---|---|---|---|---|---|
| | F1-score | Precision | Recall | F1-score | Precision | Recall |
| P1 | 0.43 | 0.52 | 0.37 | **0.63** | **0.85** | **0.50** |
| P2 | 0.47 | 0.63 | 0.38 | **0.72** | **0.90** | **0.67** |
| P3 | 0.64 | 0.60 | 0.70 | **0.80** | **0.82** | **0.83** |
| P4 | 0.46 | 0.65 | 0.37 | **0.85** | **0.93** | **0.80** |
| P5 | 0.45 | 0.45 | 0.45 | **0.78** | **0.90** | **0.69** |
| P6 | 0.61 | 0.75 | 0.55 | **0.84** | **0.98** | **0.75** |
| P7 | 0.40 | 0.46 | 0.35 | **0.66** | **0.91** | **0.54** |
| P8 | 0.51 | 0.75 | 0.42 | **0.55** | **0.70** | **0.46** |
| P9 | 0.64 | 0.72 | 0.61 | **0.77** | **0.91** | **0.69** |
| P10 | 0.45 | 0.45 | 0.46 | **0.85** | **0.92** | **0.80** |
| P11 | 0.42 | 0.47 | 0.39 | **0.60** | **0.75** | **0.52** |
| P12 | 0.38 | 0.48 | 0.31 | **0.63** | **0.72** | **0.56** |
| P13 | 0.57 | 0.68 | 0.52 | **0.75** | **0.89** | **0.69** |
| P14 | 0.41 | 0.60 | 0.32 | **0.73** | **0.92** | **0.60** |
| P15 | 0.44 | 0.62 | 0.36 | **0.74** | **0.91** | **0.66** |
| P16 | 0.54 | 0.67 | 0.46 | **0.67** | **0.90** | **0.54** |
| P17 | **0.68** | 0.86 | 0.59 | 0.67 | 0.86 | 0.59 |
| P18 | **0.92** | **0.96** | **0.90** | 0.78 | 0.94 | 0.67 |

**Table 1: The performance of the model with and without labeled counterfactuals for each participant**

*[then] maybe it has not learned anything about products. It has just learned things about other things"* (i.e., model overfitting). Alternatively, P5 observed that, if the generated counterfactuals belong to both the original and target labels, the learned pattern rule might not be accurate enough (i.e., underfitting), remarking, *"[the model] seems more confident"* when the generated counterfactuals belong exclusively to the target labels.

*Highlighting alignable differences allowed participants to compare and contrast data points during annotation.* Participants found that annotating counterfactuals—that were rendered with alignable differences highlighted—helpful in their decision-making. Specifically, two patterns of decision-making emerged. First, some participants (P3, P4, P13) used the counterfactuals to revisit their interpretations of the original example and provide additional labels. For example, P4 adds a *"product"* label to a sentence they previously labeled as *"service"* after labeling the counterfactuals. In the followup interview, they reasoned that establishments that sell service as their products like a doctors office need to have the *'product'* label as well. Similarly, P3 stated, *"I think by changing different parts of it [the original sentence], it highlighted a part of the sentence that I was not previously focused on, and so that did help me sort of reframe from what I initially had labeled it."* By the end of the session, P3 reflected on how their initial interpretation of the label *'service'* had changed: *"I guess, going back to service, I did redefine, like we could also be talking about like the staff there, so like, including that in the labels."* We also see participants similarly updating their understanding and interpretations of labels after seeing the counterfactuals: *"it is more like the loose [definition of] happy we are talking about not just the word happy, but it sounds like it is just not like a vague mapping to the emotion"* (P16).

Another decision-making behavior we saw consistently, perhaps an inverse of the previous one, was the participants' reliance on their original labeling to decide the counterfactuals' labels. This meant that the participants used the original example as an anchor for their consecutive interpretations. P18 used the original learned pattern rule for the label *'fearful'* to provide an additional label for a generated counterfactual example. Specifically, after seeing that the generated counterfactual with the suggested label *'sad'* also matched the pattern rule for the label *'fearful'*, they annotated the sentence with both labels and stated, *"I did not have that context [in the previous conditions] like I set up for myself, because reading this, I have like a story scenario in my head. [If] I did not have a scenario set up for myself I [would] probably just label it as only sad right away"*. The context provided in C3 allowed participants to infer additional labels to data points; this rarely happened in C1 (Non VT counterfactuals) where participants thought the generated examples were *'independent'* (P1) from the original example and felt that they were labeling *'completely new'* (P18) data points. In C2 (VT counterfactuals without highlighted alignable differences), despite following a similar approach to C3, participants struggled to identify differences between counterfactual examples, suggesting the effectiveness of highlighting structural consistency and differences in aiding comparative analysis.

## 5 Discussion and Design Implications

The results from our user study suggest that both the participants and the model benefited from the Variation Theory (VT)-based counterfactuals and Structural Alignment Theory (SAT)-based rendering. Participants were able to efficiently focus on key differences between the original and counterfactual examples, which facilitated more efficient annotations. The participants also found

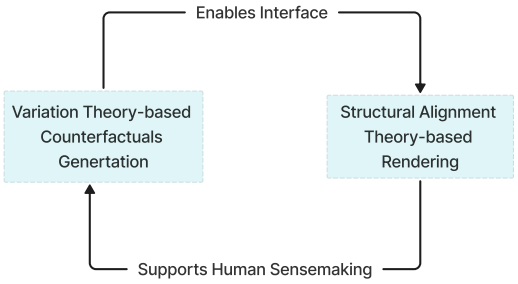


**Figure 6: Our study finds a bilateral relationship between Variation Theory and Structural Alignment Theory. The Variation Theory-based counterfactual generation method enabled the rendering of structurally alignable differences. In turn, the rendering supported the users' sensemaking of the variation in the generated counterfactuals**

Mocha to be useful in helping them refine and evolve their label definitions while giving them insight into the behavior of the model. Although the benefits of only Variation Theory-based counterfactuals (without SAT-based rendering) were not immediately evident in participants' experience, they were critical in enabling SAT-based rendering, which was found to be effective (Fig. 6). Below we further discuss our findings and design implications.

## 5.1 A Symbiotic Relationship Between Variation Theory and Structural Alignment Theory

The results of our study indicate that participants spent significantly less time annotating batches of counterfactuals when they were rendered according to SAT compared to other conditions i.e., supporting the participants' selective focus on the varying phrases, rather than phrases that stay consistent. Notably, we found no statistically significant differences in annotation time between C2 (VT-based counterfactual without SAT) and the baseline condition in C1. Although the counterfactuals in both C2 and C3 were based on Variation Theory, the participants' annotation efficiency was particularly enhanced by the SAT-based rendering method introduced in C3. This finding is consistent with previous work that supports users' sense-making of text, e.g., by modulating text saliency. Specifically, Gu et al. [32] and Gero et al. [29] both found improved reading efficiency and comprehension with saliency-modulating text renderings. While SAT-based rendering supported human sensemaking in both Gero et al. [29] and Mocha, we also show that the combination of VT and SAT support the model's learning.

To support a user's cognitive process for comparison [62], the version of Mocha (C3) that followed a combination of Variation Theory and Structural Alignment Theory was consistently more effective. We argue that these two theories form a symbiotic relationship (Fig. 6). Variation Theory provides the conceptual basis for generating structurally consistent differences, while Structural Alignment Theory (SAT) enhances the user's ability in recognizing and processing these differences. This symbiotic relationship stems from the fact that Structural Alignment Theory (SAT) enhances the salience of differences, while the way we used Variation Theory (VT) to generate contradicting examples across the boundaries of labels ensures that these differences are conceptually informative. By helping users see alignable differences, SAT-based rendering helps users focus on key variations that are essential to changing the data item's label, making it easier to interpret the effects of changes and their significance. Thus, the integration of both theories enables users to efficiently process and compare variations, leading to more informed decisions and a clearer understanding of the model's behavior.

## 5.2 Human Cognition and Learning Theories in the Interactive ML Pipeline

In its design, Mocha controls both how counterfactual data is generated and how they are rendered to the user. By incorporating theories such as Structural Alignment Theory and Variation Theory, it aims to support the learning of both the human and the model. These theories have proven insightful for understanding how humans grasp and compare concepts, shaping the development of human-AI collaboration systems for sensemaking [29], hypothesis testing [2], as well as model training [24]. From a human-in-the-loop machine learning perspective, Mocha addresses two seemingly contradictory objectives: (1) generating labeled data that diversifies the training dataset to aid the model's learning, and (2) maintaining structural consistency across the batches of data presented to users to support their cognitive processes. Mocha achieves this balance by enforcing a common structure through the model's learned pattern rules [25]. By visualizing these consistent pattern rules, users may be better understanding the behavior of the model through inference projection [26]. This can not only boosts the model's performance but also enable participants to validate or correct the model during the interactive training process.

Although visual cues for alignable differences in Mocha were helpful in supporting the participants' reasoning, Estes and Hasson [17] argue that while alignable differences can be more straightforward and easier for comparison, non-alignable differences can also provide key information that might otherwise remain overlooked. These differences necessitate a more abstract form of comparison, prompting users to think beyond simple relational structures and consider broader conceptual frameworks. For example, when comparing planes and cars, their alignable differences can be that they both have engines, but the engines are different in size. While this gives us some insight into their definitions, comparing a plane's wings and a car's wheels, which are not structurally alignable but conceptually and analogically alignable, gives additional insight to categorizing one for land and the other for air. Future research should explore how non-alignable differences in AI explanations affect user decision-making and understanding. Such studies could determine whether these non-alignable comparisons enhance user performance and elicit deeper insights in human-AI collaborative systems.

# 6 Limitations and Future Work

The current design of the user study, which allowed participants to interact with each condition for 25 minutes, yielded valuable insights into their immediate reactions and interactions. However,

a longer study duration would provide a deeper understanding of how users engage with the system throughout various stages of the model's learning. Extending the study period would enable observations on how users' strategies evolve as the model improves, potentially uncovering significant insights about the long-term dynamics of human-AI collaboration, especially in relation to trust building and mental model refinement. Our study also only used example data from two domains, providing limited evidence to the approach's ecological validity in other data domains. In future work, we aim to investigate longer-term interactions of users in diverse application domains through deployment studies to uncover dynamic patterns of collaboration.

We also tested our system exclusively with the training of a neuro-symbolic model. While this serves as a compelling use case, it would be beneficial to explore the effectiveness of our approach with different types of models, such as purely statistical machine learning models, deep learning architectures, or hybrid systems, to better understand the generalizability of our approach across different paradigms. Different models might offer varying challenges and affordances in terms of explainability, interaction transparency, and feedback responsiveness. Exploring these dimensions would provide more insight into our proposed approach.

## 7 Conclusion

This paper introduced Mocha, an interactive machine learning tool informed by two theories of human concept learning and cognition. Based on the Variation Theory of human learning, it generates synthetic counterfactual data that are syntactically and semantically similar to already-annotated data but predicted by pre-trained large language models to have different labels. Following Structural Alignment Theory, it renders the generated counterfactuals aligned in batches with differences and similarities highlighted to support the user's cognitive process of interpreting and understanding data. A lab study with 18 participants demonstrated the usability of Mocha and its effectiveness in improving annotation efficiency, enhance the model's learning, and facilitating co-adaptive learning in which users gain insight into the state of the model and reflect on their own understanding. Mocha exemplified the application of human cognition and concept learning theories in the interactive machine learning pipeline to support the negotiation of conceptual boundaries for bi-directional human-AI alignment.

## Acknowledgments

This work was supported in part by an IBM Ph.D. Fellowship, an AnalytiXIN Faculty Fellowship, a Notre Dame-IBM Technology Ethics Lab Award, a Google Research Scholar Award, Alfred P. Sloan Foundation FG-2023-19960, and NSF Grants IIS-2107391, CCF-2123965, CMMI-2326378, and CNS-2426395. Any opinions, findings, or recommendations expressed here are those of the authors and do not necessarily reflect the views of the sponsors.

## References

[1] Ebba Cecilia Ovesdotter Alm. 2008. *Affect in* text and speech.* University of Illinois at Urbana-Champaign.

[2] Ian Arawjo, Chelse Swoopes, Priyan Vaithilingam, Martin Wattenberg, and Elena L Glassman. 2024. ChainForge: A Visual Toolkit for Prompt Engineering and LLM Hypothesis Testing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–18.

[3] Shilpa Arora and Sachin Agarwal. 2007. Active learning for natural language processing. *Language Technologies Institute School of Computer Science Carnegie Mellon University* 2 (2007).

[4] Lora Aroyo, Matthew Lease, Praveen Paritosh, and Mike Schaekermann. 2022. Data excellence for AI: why should you care? *Interactions* 29, 2 (2022), 66–69.

[5] Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362* (2016).

[6] Serena Booth, Sanjana Sharma, Sarah Chung, Julie Shah, and Elena L Glassman. 2022. Revisiting human-robot teaching and learning through the lens of human concept learning. In *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 147–156.

[7] Michael Brooks, Saleema Amershi, Bongshin Lee, Steven M Drucker, Ashish Kapoor, and Patrice Simard. 2015. FeatureInsight: Visual support for error-driven feature ideation in text classification. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 105–112.

[8] Niklas Bunzel, Nicolas Göller, and Raphael Antonius Frick. 2024. Identifying and Generating Edge Cases. In *Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems*. 16–23.

[9] Ángel Alexander Cabrera, Marco Tulio Ribeiro, Bongshin Lee, Robert Deline, Adam Perer, and Steven M Drucker. 2023. What did my AI learn? How data scientists make sense of model behavior. *ACM Transactions on Computer-Human Interaction* 30, 1 (2023), 1–27.

[10] Joseph Chee Chang, Aniket Kittur, and Nathan Hahn. 2016. Alloy: Clustering with crowds and computation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3180–3191.

[11] Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2022. DISCO: Distilling counterfactuals with large language models. *arXiv preprint arXiv:2212.10534* (2022).

[12] Isabel F Cruz and William Sunna. 2008. Structural alignment methods with applications to geospatial ontologies. *Transactions in GIS* 12, 6 (2008), 683–711.

[13] Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *AAAI*, Vol. 5. 746–751.

[14] Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. CORE: A retrieve-then-edit framework for counterfactual data generation. *arXiv preprint arXiv:2210.04873* (2022).

[15] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305* (2020).

[16] John J Dudley and Per Ola Kristensson. 2018. A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 8, 2 (2018), 1–37.

[17] Zachary Estes and Uri Hasson. 2004. The importance of being nonalignable: a critical test of the structural alignment theory of similarity. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 30, 5 (2004), 1082.

[18] Jerry Alan Fails and Dan R Olsen Jr. 2003. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*. 39–45.

[19] Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. Causalm: Causal model explanation through counterfactual language models. *Computational Linguistics* 47, 2 (2021), 333–386.

[20] Richard M Felder and Rebecca Brent. 2009. Active learning: An introduction. *ASQ higher education brief* 2, 4 (2009), 1–5.

[21] Andy P Field. 2005. Kendall's coefficient of concordance. *Encyclopedia of statistics in behavioral science* 2 (2005), 1010–11.

[22] Joseph L Fleiss, Jacob Cohen, and Brian S Everitt. 1969. Large sample standard errors of kappa and weighted kappa. *Psychological bulletin* 72, 5 (1969), 323.

[23] Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association* 32, 200 (1937), 675–701.

[24] Simret Araya Gebreegziabher, Kuangshi Ai, Zheng Zhang, Elena L. Glassman, and Toby Jia-Jun Li. 2024. Leveraging Variation Theory in Counterfactual Data Augmentation for Optimized Active Learning. *arXiv preprint arXiv:2408.03819* (2024).

[25] Simret Araya Gebreegziabher, Zheng Zhang, Xiaohang Tang, Yihao Meng, Elena L Glassman, and Toby Jia-Jun Li. 2023. Patat: Human-ai collaborative qualitative coding with explainable interactive rule synthesis. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–19.

[26] Dedre Gentner. 2010. Bootstrapping the mind: Analogical processes and symbol systems. *Cognitive science* 34, 5 (2010), 752–775.

[27] Dedre Gentner and Virginia Gunn. 2001. Structural alignment facilitates the noticing of differences. *Memory & cognition* 29, 4 (2001), 565–577.

[28] Dedre Gentner and Arthur B Markman. 1997. Structure mapping in analogy and similarity. *American psychologist* 52, 1 (1997), 45.

[29] Katy Ilonka Gero, Chelse Swoopes, Ziwei Gu, Jonathan K Kummerfeld, and Elena L Glassman. 2024. Supporting Sensemaking of Large Language Model Outputs at Scale. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–21.

[30] Marco Gillies, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin

Lee, et al. 2016. Human-centred machine learning. In *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems*. 3558–3565.
[31] Inês Gomes, Luís F Teixeira, Jan N Van Rijn, Carlos Soares, André Restivo, Luís Cunha, and Moisés Santos. 2024. Finding Patterns in Ambiguity: Interpretable Stress Testing in the Decision Boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8316–8321.
[32] Ziwei Gu, Ian Arawjo, Kenneth Li, Jonathan K Kummerfeld, and Elena L Glassman. 2024. An AI-Resilient Text Rendering Technique for Reading and Skimming Documents. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–22.
[33] Omeed Habibelahian, Rajesh Shrestha, Arash Termehchy, and Paolo Papotti. 2022. Exploratory training: when trainers learn. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. 1–5.
[34] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.
[35] Sabit Hassan and Malihe Alikhani. 2023. D-CALM: A dynamic clustering-based active learning approach for mitigating bias. *arXiv preprint arXiv:2305.17013* (2023).
[36] Marasović Jacovi. 2021. Formalizing Trust in Artificial Intelligence: Prerequisites, Causes and Goals of Human Trust in AI. *URL https://dl.acm.org/doi/10.1145/3442188.3445923* (2021).
[37] Hannu Kivijärvi. 2018. Advancing Organizational Alignment Decisions: Insights from the Structural Alignment Theory to the Business-IT Alignment Problem. *International Journal of IT/Business Alignment and Governance (IJITBAG)* 9, 1 (2018), 53–80.
[38] Carmen Konzett-Firth. 2020. Co-adaptation processes in plenary teacher-student talk and the development of L2 interactional competence. *Classroom Discourse* 11, 3 (2020), 209–228.
[39] William H Kruskal and W Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association* 47, 260 (1952), 583–621.
[40] Todd Kulesza, Saleema Amershi, Rich Caruana, Danyel Fisher, and Denis Charles. 2014. Structured labeling for facilitating concept evolution in machine learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3075–3084.
[41] Chi-Chun Lee, Athanasios Katsamanis, Panayiotis G Georgiou, and Shrikanth S Narayanan. 2012. Based on Isolated Saliency or Causal Integration? Toward a Better Understanding of Human Annotation Process using Multiple Instance Learning and Sequential Probability Ratio Test.. In *INTERSPEECH*. 619–622.
[42] David D Lewis. 1995. A sequential algorithm for training text classifiers: Corrigendum and additional data. In *Acm Sigir Forum*, Vol. 29. ACM New York, NY, USA, 13–19.
[43] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764* (2021).
[44] Ference Marton. 2014. *Necessary conditions of learning*. Routledge.
[45] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica* 22, 3 (2012), 276–282.
[46] Robert Munro Monarch. 2021. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster.
[47] Markus Neuhäuser. 2011. Wilcoxon-Mann-Whitney Test.
[48] Alva Noë. 2022. *Learning to Look: Dispatches from the Art World*. Oxford University Press.
[49] Husam Quteineh, Spyridon Samothrakis, and Richard Sutcliffe. 2020. Textual data augmentation for efficient active learning on tiny datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7400–7410.
[50] Gonzalo Ramos, Christopher Meek, Patrice Simard, Jina Suh, and Soroush Ghorashi. 2020. Interactive machine teaching: a human-centered approach to building machine-learned models. *Human–Computer Interaction* 35, 5-6 (2020), 413–451.
[51] Marcel Robeer, Floris Bex, and Ad Feelders. 2021. Generating realistic natural language counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. 3611–3625.
[52] Raphael Schumann and Ines Rehbein. 2019. Active learning via membership query synthesis for semi-supervised sentence classification. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*. 472–481.
[53] Burr Settles. 2009. Active learning literature survey. (2009).
[54] Knearem shen. 2024. Towards Bidirectional Human-AI Alignment: A Systematic Review for Clarifications, Framework, and Future Directions. *URL http://arxiv.org/abs/2406.09264* (2024).
[55] Rajesh Shrestha, Omeed Habibelahian, Arash Termehchy, and Paolo Papotti. 2023. Exploratory Training: When Annonators Learn About Data. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–25.
[56] Jina Suh, Soroush Ghorashi, Gonzalo Ramos, Nan-Chen Chen, Steven Drucker, Johan Verwey, and Patrice Simard. 2019. Anchorviz: Facilitating semantic data exploration and concept discovery for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 10, 1 (2019), 1–38.
[57] Jeffrey M Rzeszotarski Swati Mishra. 2023. Human Expectations and Perceptions of Learning in Machine Teaching. (2023), 13–24.
[58] Annalisa Szymanski, Simret Araya Gebreegziabher, Oghenemaro Anuyah, Ronald A Metoyer, and Toby Jia-Jun Li. 2024. Comparing Criteria Development Across Domain Experts, Lay Users, and Models in Large Language Model Evaluation. *arXiv preprint arXiv:2410.02054* (2024).
[59] Annalisa Szymanski, Noah Ziems, Heather A Eicher-Miller, Toby Jia-Jun Li, Meng Jiang, and Ronald A Metoyer. 2024. Limitations of the LLM-as-a-Judge Approach for Evaluating LLM Outputs in Expert Knowledge Tasks. *arXiv preprint arXiv:2410.20266* (2024).
[60] Karan Taneja, Harshvardhan Sikka, and Ashok Goel. 2022. Human-AI Interaction Design in Machine Teaching. *arXiv preprint arXiv:2206.05182* (2022).
[61] Michael Williams and Tami Moser. 2019. The art of coding and thematic exploration in qualitative research. *International management review* 15, 1 (2019), 45–55.
[62] Edward J Wisniewski and Miriam Bassok. 1999. What makes a man similar to a tie? Stimulus compatibility with comparison and integration. *Cognitive psychology* 39, 3-4 (1999), 208–238.
[63] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. *arXiv preprint arXiv:2101.00288* (2021).
[64] Litao Yan, Miryung Kim, Bjoern Hartmann, Tianyi Zhang, and Elena L Glassman. 2022. Concept-annotated examples for library comparison. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–16.
[65] Yazhou Yang and Marco Loog. 2016. Active learning using uncertainty information. In *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2646–2651.
[66] Pengyi Zhang and Dagobert Soergel. 2016. Process patterns and conceptual changes in knowledge representations during information seeking and sensemaking: A qualitative user study. *Journal of Information Science* 42, 1 (2016), 59–78.
[67] Shengnan Zhang, Yan Hu, and Guangrong Bian. 2017. Research on string similarity algorithm based on Levenshtein Distance. In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. IEEE, 2247–2251.
[68] Zheng Zhang, Zheng Ning, Chenliang Xu, Yapeng Tian, and Toby Jia-Jun Li. 2023. PEANUT: A Human-AI Collaborative Tool for Annotating Audio-Visual Data. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–18.
[69] Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. A survey of active learning for natural language processing. *arXiv preprint arXiv:2210.10109* (2022).

# A Appendix

## A.1 User Study Participant Data

| PID | Level of Education | Age | Gender | ML Experience |
|---|---|---|---|---|
| P1 | Bachelor's | 18-24 | Male | Beginner |
| P2 | Bachelor's | 18-24 | Male | Expert |
| P3 | Bachelor's | 18-24 | Male | Expert |
| P4 | PhD | 25-34 | Female | Intermediate |
| P5 | Bachelor's | 25-34 | Male | Intermediate |
| P6 | Master's | 25-34 | Male | Expert |
| P7 | Master's | 25-34 | Male | Intermediate |
| P8 | Master's | 25-34 | Male | Intermediate |
| P9 | Master's | 25-34 | Male | Intermediate |
| P10 | Master's | 25-34 | Male | Intermediate |
| P11 | Master's | 18-24 | Male | Intermediate |
| P12 | Master's | 25-34 | Female | Beginner |
| P13 | Bachelor's | 18-24 | Female | Intermediate |
| P14 | Master's | 25-34 | Female | None |
| P15 | Bachelor's | 18-24 | Female | Intermediate |
| P16 | Master's | 18-24 | Male | Beginner |
| P17 | Master's | 25-34 | Female | Intermediate |
| P18 | Bachelor's | 18-24 | Female | Beginner |

**Table 2: Participant demographic data**

## A.2 User Study Participant Task Details

| Participant ID | Data | Condition Order |
|---|---|---|
| P1 | Yelp | C1 - C3 - C2 |
| P2 | Yelp | C2 - C3 - C1 |
| P3 | Yelp | C3 - C1 - C2 |
| P4 | Yelp | C1 - C2 - C3 |
| P5 | Yelp | C1 - C3 - C2 |
| P6 | Yelp | C2 - C1 - C3 |
| P7 | Yelp | C3 - C2 - C1 |
| P8 | Yelp | C1 - C3 - C2 |
| P9 | Yelp | C3 - C1 - C2 |
| P10 | Emotions | C1 - C3- C2 |
| P11 | Emotions | C3 - C1 - C2 |
| P12 | Emotions | C2 - C1 - C3 |
| P13 | Emotions | C2 - C3 - C1 |
| P14 | Emotions | C3 - C1 - C2 |
| P15 | Emotions | C3 - C2 - C1 |
| P16 | Emotions | C1 - C2 - C3 |
| P17 | Emotions | C1 - C3 - C2 |
| P18 | Emotions | C2 - C1 - C3 |

**Table 3: User study participant's assigned data and condition order**

## A.3 Neuro-symbolic Pattern Rules

The neuro-symbolic model adopts an iterative learning approach to delineate the boundaries of concepts represented by data points and their corresponding ground truth labels. Although the current simulation study employs ground truth labels, these will eventually be replaced with human annotations in future interactive systems. Following a random selection of a subset of annotations, the interactive program synthesis method from PaTAT [25] is applied to derive domain-specific pattern rules that align with the annotated examples. These rules capture the lexical, syntactic, and semantic similarities present among data sharing the same label. The pattern language is composed of the following components:

- Part-of-speech (POS) tags: `VERB`, `PROPN`, `NOUN`, `ADJ`, `ADV`, `AUX`, `PRON`, `NUM`
- Word stemming: `[WORD]` (e.g., `[have]` will match all variants of have, such as *had*, *has*, and *having*)
- Soft match: `(word)` (e.g., `(pricey)` will match synonyms such as *expensive* and *costly*, etc.)
- Entity type: `$ENT-TYPE` (e.g., `$LOCATION` will match phrases of location type, such as *Houston, TX* and *California*; `$DATE` will match dates; `$ORG` will match names of organizations)
- Wildcard: `*` (will match any sequence of words)

## A.4 Candidate Phrase Generation Prompt

```
The assistant will create a list of candidate
    phrases that match the given symbolic domain
    specific pattern. The domain specific pattern
    definition is given below. The domain specific
    pattern symbols includes the following patterns:
Part-of-speech (POS) tags are capital: VERB, PROPN,
    NOUN, ADJ, ADV, AUX, PRON, NUM
Word stemming are surrounded in [] and should have
    an exact match: [WORD] (e.g., [have] will match
    all variants of have)
Soft match are surrounded by () and will match words
    with their synonyms. The list of synonms for
    each soft match in a pattern are given in the
    user instruction: (word) (word will only be
    matched with a limited set of similar words
    provided in this instruction)
Entity type start with $ sign: $ENT-TYPE (e.g.,
    $LOCATION will match phrases of location type,
    such as Houston; $DATE will match dates)
Wildcard is the * symbol and can match anything: *
    (will match any sequence of words)
The patterns can be combined using an and operator
    (+) or an or operator (|). For example the
    pattern 'VERB␣+␣PROPN' will match any sentence
    that has a verb followed by a proper noun. The
    pattern VERB|PROPN will match anything that is a
    verb or a proper noun.
Soft matches can only be replaced with a list of
    available words.
```

```
For the following text and pattern, generate as many
    diverse example phrases that match the given
    pattern and can be part of the given target
    label. Separated your answer by a comma
# Example input:
# sentence: 'Too many other places to shop with
    better prices .'
# phrase to modify: 'prices .'
# pattern: '(price)+*'
# current label: price
# softmatch: [price:[purchase, pricey, cheap, cost,
    pricing]]
# target label: service
# Example output:
# [purchase options, pricey service, cheap help,
    pricing plans, cost breakdown]}
```

## A.5 Counterfactual Generation Prompt

```
Your task is to modify a given sentence so that it
    aligns with a target label instead of its
    original label, making only necessary changes.
    Follow these steps:
- Generate Target Phrases: Identify phrases relevant
    to the target label within the context of the
    original sentence.
- Modify the Sentence: Use one of the generated
    target phrases to adjust the original sentence,
    ensuring that:
- The modified sentence no longer fits the original
    label and does not reference or imply the
    original label in any way.
- The modified sentence is appropriate for the
    target label and logically coherent.
- The modified sentence should be natural and
    fluent, making sense as a standalone sentence.
- Changes made are necessary while keeping the
    original sentence structure as intact as
    possible. To preserve the quality of the new
    sentence you can remove or add necessary parts.
- The modification includes one of the provided
    candidate phrases, replacing the highlighted
    portion of the original sentence.
- Explanation (Optional): If necessary, provide a
    brief explanation of why the modified sentence
    fits the target label.
Explanation of Terms:
* phrase about target label: generated phrases
    relevant to the target label that help guide the
    sentence modification.
* phrases to include: this includes one phrase from
    'phrase␣about␣target␣label' and another phrase
    from the user input 'candidate␣phrases'. these
    two phrases will be incorporated into the
    modified sentence.
* modified sentence: The final sentence after
    modification to align with the target label. It
    should be natural, logical, and coherent.
* reason: A brief explanation of why the modified
    sentence fits the target label.
* label: The final label assigned to the modified
    sentence, reflecting its changes.
* Ensure the final sentence is correctly labeled
    according to the target label, with no
    references to the original label, and with
    minimal deviation from the original content.
# Example input:
# Original sentence: 'The wings were delicious.'
# Original label: product,
# Target label: price,
# Candidate phrases: ['yummy', 'tasty', 'flavour',
    'deliciousness', 'taste', 'delicious']
# phrase about target label: ['cheap', 'expensive',
    'pricey']
# phrase to include: ['taste', 'cheap']
# Example output:
# modified sentence: 'The wings were cheap for the
    taste.'
# reason: 'The sentence shifts focus to the cost of
    the wings, making it fit the target label price.'
# label: price
```

## A.6 NASA-TLX Results

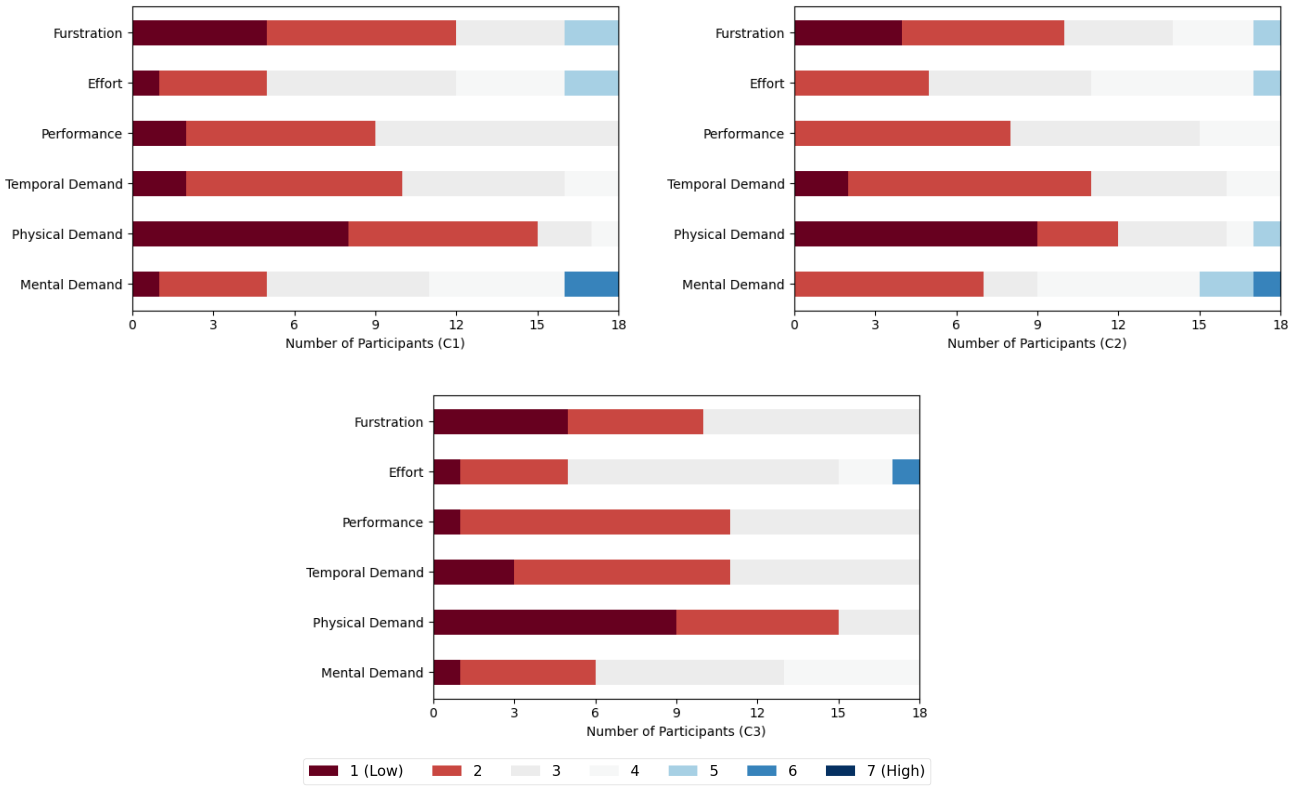


**Figure 7: NASA-TLX results from the user study**