ABSTRACT
With ever increasing bodies of unlabeled data from complex, dynamic and weakly structured domains, ML is more necessary than ever. Yet domain experts find it difficult to understand most ML algorithms, and so cannot easily retrain them as new data arrives. Interactive machine learning techniques have been proposed to take advantage of human's ability to categorize in these complex domains, but little attention has been paid to building interfaces for non-ML experts to provide input, and in particular to creating a user experience that engenders trust.

Qualitative coding - the decades-old practice of manual classification - provides a proven methodology that can be adapted to structure interaction between domain experts and ML algorithms. Here we explore how an immersive system can be built to leverage QC’s intuitive methods and grow a trusting partnership between human and ML classifiers.

INTRODUCTION
As the amount of data we generate increases, the need for computer-aided analysis grows.

For end user, tuning a robust ML model generally needs expert assistance, since such tuning normally requires understanding the ML algorithm's inner workings. While machine learners perform well in clearly defined domains, there are still many complex domains where available data sets do not fully enable ML because they may be incomplete or uncertain.

The use of large displays for visual analytics in large datasets has been proposed to better support sense-making via interactive visual exploration and a more natural interface than traditional interface elements, documents. While this focuses on providing easier interaction for domain experts, as yet, there's nothing that focuses on classification rather than clustering.

To address this lack, we propose structuring this dialogue with the methods of Qualitative Coding (QC).

QUALITATIVE CODING
Qualitative coding is a manual classification method commonly employed in the humanities and behavioral sciences used to extract meaning from non-numeric data such as text, imagery, and video. Using grounded theory, labels are often created as needed by coders, rather than working with a set of labels defined rigidly before beginning.

Codebook – index of labels used in first cycle
Analytic Memos - attached to data to explain reason for choosing
Data displays – blanket term referring to any method of organization that coders use to visually summarize their work.
History Log – to store all data, since manual coding can’t be undone

Our goal is to increase the utility and accessibility of ML algorithms by making interaction with them understandable and efficient enough to allow domain experts without ML expertise to train them, and to explain their results to their peers.

QC-ML INTERACTION: SOLUTION PROPOSED
We aim to solve following challenges faced by Human-ML interaction, through QC:
- Recall
- Collaboration
- Error Correction
- Iteration
- Efficiency

We believe the ideal ML UI will require three key elements: (Fig. 2)

Model Visualization – An interactively updated visual summarizing how model organizes data overall. This summary helps users monitor their labeling/coding progress as they work with a dataset that is much larger than in traditional QC.

Derived Data – Includes codebook and history, in order to show most frequently used labels.

Data Display – A zoomed-in view corresponding most directly to QC's data display. When a user selects a slice in model visualization (refer figure), data display shows significant items in that slice, ordered and color-coded.

Partnered QC depends on fostering a dialogue between coders, and in our scenario, we consider the underlying ML system to be a partner in the coding effort. Users have the freedom to explore the data at will, but the system - as the ML expert - needs to direct the labeling to foster the creation of a robust model. A purely virtual version of this interface could approximate the analogue immersive environments used by QC experts such as a tabletop with digital post-its.

DISCUSSION
One primary goal of the system is to (ideally) eliminate the need for an ML expert to assist during the building and training of a model. The other major consideration in building this type of system is whether or not users will take advantage of this new capability. Further goals for the system include group collaboration and inclusion of and consideration for non-domain experts as collaborators. Our proposed QC/ML system should fulfill several of these heuristics. System status should always be visible via the codebook and the data display. The system assists with error recovery by maintaining a history supporting undo, and automatically evaluating feedback from the human users and highlighting poorly fitting data for further consideration. QC-based language and structure of the system should be easily understood by users, as it minimizes references to ML terminology, technology and components.