Kaleidoscope: Semantically-grounded, Context-specific ML Model Evaluation

Kaleidoscope: Semantically-grounded, context-specific ML model evaluation

Harini Suresh
hsuresh@mit.edu
MIT CSAIL
USA

Divya Shanmugam
divyas@mit.edu
MIT CSAIL
USA

Tiffany Chen
tiffc@mit.edu
MIT CSAIL
USA

Annie Bryan
annieb22@mit.edu
MIT CSAIL
USA

Alexander D'Amour
alexdamour@google.com
Google Research
USA

John V. Guttag
guttag@mit.edu
MIT CSAIL
USA

Arvind Satyanarayan
arvindsatya@mit.edu
MIT CSAIL
USA

ABSTRACT

Desired model behavior often differs across contexts (e.g., different geographies, communities, or institutions), but there is little infrastructure to facilitate context-specific evaluations key to deployment decisions and building trust. Here, we present Kaleidoscope, a system for evaluating models in terms of user-driven, domain-relevant concepts. Kaleidoscope's iterative workflow enables generalizing from a few examples into a larger, diverse set representing an important concept. These example sets can be used to test model outputs or shifts in model behavior in semantically-meaningful ways. For instance, we might construct a "xenophobic comments" set and test that its examples are more likely to be flagged by a content moderation model than a "civil discussion" set. To evaluate Kaleidoscope, we compare it against template- and DSL-based grouping methods, and conduct a usability study with 13 Reddit users testing a content moderation model. We find that Kaleidoscope facilitates iterative, exploratory hypothesis testing across diverse, conceptually-meaningful example sets.

ACM Reference Format:
Harini Suresh, Divya Shanmugam, Tiffany Chen, Annie Bryan, Alexander D'Amour, John V. Guttag, and Arvind Satyanarayan. 2023. Kaleidoscope: Semantically-grounded, context-specific ML model evaluation. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3544548.3581482

1 INTRODUCTION

There is increasing recognition that evaluations of machine learning (ML) systems should be grounded in *context* [25, 41]. Context is a broad term, and might denote geographies, physical or virtual communities, organizations, institutions, or more. Stakeholders in different contexts have different lived experiences, goals, and notions of what constitutes a "model failure" [11, 12, 23, 30, 48].

Consider the example of content moderation, which we use as a running case study throughout this paper. Different online communities deal with different types of harassment or trolling, and enforce a wide range of rules/norms [7, 15]. For instance, some subreddits ban talking about specific topics, such as guns or diet advice, while others do not. Specific phrases or emojis might appear to be offensive in one context, but correspond to inside jokes or meanings in another [38]. As automated moderation tools become increasingly available [22, 31, 54], how can the users and/or moderators of an online community understand where a particular system succeeds/fails for them, and assess whether it is suited to their context?

There is currently little infrastructure for users in a particular context to pose or begin to answer these questions. Standard evaluations on static benchmark datasets are often misaligned with real-world deployment contexts, due to issues such as distribution shift [40, 45], underspecification [9], or poor subgroup performance [3]. Evaluations that focus on specific subgroups [35], other metrics [20, 37], or new benchmark datasets [26, 28, 56] are also limited, since they are tied to predefined subgroup labels or metrics. Other work has taken a more bespoke approach, compiling context-specific evaluation datasets from scratch with specific groups of users [44, 50], or proposing complex causal models of societal context from first principles [33]. The resulting evaluations are valuable, but they require significant time, effort, and customization to design (or to update, if/when user needs evolve over time).

To address this gap, we present Kaleidoscope, a workflow and interactive user interface for performing user-driven, context-specific evaluations of ML models. Kaleidoscope leverages users' implicit expectations of "good model behavior" in a given context, and helps them translate these behaviors into explicitly defined tests.

Using Kaleidoscope's iterative workflow, users *identify* important examples using data from their own context, *generalize* them

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3581482

into semantically-meaningful concepts, and *specify and test* model behavior on those concepts. This workflow enables a bottom-up approach, where users can start with a few examples of a particular concept and generalize them into a large, representative *example set* by adding semantically-similar examples retrieved in a learned embedding space. The process is designed to be iterative and exploratory—rather than requiring a precise definition of the concept upfront, its bounds can become more complete and precise as users find and add new examples.

Users can then specify and evaluate model behavior on these example sets by defining and running *tests*. We distill two axes to specify model behaviors: the behavior type (e.g., specific model outputs, invariances, or shifts) and its granularity (e.g., whether it pertains to a single example set, aggregate comparisons between two example sets, or pairwise comparisons after applying a transformation to each example in a set). Specifying tests makes desired model behaviors transparent, and running them surfaces insights into model strengths and limitations in terms of domain-relevant concepts. In doing so, tests can build trust by making anticipated behaviors explicit (i.e., facilitating *contractual trust* [25]).

To evaluate Kaleidoscope, we conduct a two-part evaluation. First, using the Cognitive Dimensions of Notation heuristic framework [19], we contrast Kaleidoscope's conceptual affordances against template-based and domain specific language (DSL-based) grouping methods for natural language tasks to better understand their tradeoffs. We find that Kaleidoscope results in more semantically-meaningful examples and tests, as opposed to lower level or syntactically focused tests. In addition, other methods require formally defining slices of data upfront. Instead, Kaleidoscope allows users to switch between exploratory and confirmatory analyses, creating slices of data that would have been difficult to define *a priori*.

We also conducted a user study with 13 Reddit users/moderators who used the system to assess two pretrained ML models for content moderation. The iterative process of finding and adding similar examples to build example sets was intuitive, and helped draw out participants' personal knowledge of the context. Participants typically started with an idea of a concept they intended to represent in an example set, but as they found and added examples, this idea sometimes expanded (as they discovered new phrases to search for or types of examples to add), became more precisely defined (as they began to delineate which similar examples did and did not belong) or split into multiple concepts (as they realized implicit subgroups within their initial idea). Resulting example sets represented concepts, drawn from personal experience or specific subreddit rules, that participants considered important (e.g., "LGBT attacks," "colorism," "disrespectful comments"). Each contained diverse examples that would be difficult or impossible to specify via templates or a DSL. Tests built off of these concepts revealed insights into model behavior that helped participants reason about if the model would work well in their context, and how it should be used.

Kaleidoscope contributes to a growing body of work that aims to gives users the agency to probe automated systems. In particular, the system helps users translate their implicit expectations of model behavior into concrete, domain-relevant tests. Our results indicate that Kaleidoscope facilitates meaningful insights into model behavior, and suggest promising directions for future work on context-grounded model evaluation.

2 RELATED WORK

Several examples have demonstrated that overall performance metrics for predictive models using static test sets do not guarantee desirable behavior in deployment, due to issues such as distribution shift [40, 45], shortcut learning [16], underspecification [9], or poor subgroup performance [3]. In response, some work has proposed documenting performance across more granular data subgroups [8, 26, 35, 56]. Typically, these include predefined demographic subgroups such as race, gender, or age. Other work has proposed a range of different evaluation metrics—e.g., notions of fairness, robustness to noise/corruptions, miscalibration, privacy, or the presence of undesirable learned correlations [9, 20, 28, 32, 37, 43, 46, 52].

Importantly, these evaluation paradigms are typically aimed at developers, and rely on several assumptions. First, they assume *a priori* definitions of success. For example, D'Amour et al. perform a number of stress tests that measure metrics outside of accuracy [9]. However, these require customized datasets and specific, predefined tasks (e.g., testing a model's robustness to corruptions with ImageNet-C [24]). Second, they assume access to static, labeled subgroups. In many cases, however, the types of examples users in a particular context care about comprise higher-level concepts [34] that are not already labeled in the data (e.g., x-rays with tricky diagnoses [4], aggressive comments [7], arrythmias with broad QRS spikes [49]). Identifying these sets of examples manually is difficult and time-consuming. And finally, they assume that desired model behavior is consistent across different deployment contexts. As is increasingly recognized, though, expectations and norms can differ widely across stakeholders and contexts (e.g., a comment considered aggressive in one community might be fine in another [30, 38]).

Some recent work has tried to address these issues and perform more context-grounded evaluations of ML systems by designing application-specific evaluation metrics or datasets with specific groups of users [38, 44, 50]. The resulting evaluations are valuable, but their design is highly bespoke. Without a guiding framework or surrounding infrastructure, redesigning this process from scratch in different contexts (or updating it for existing contexts, if user needs evolve over time) requires significant time and effort. Kaleidoscope helps fill this gap, providing a workflow and interactive system that can support context-specific evaluations.

Other work has similarly proposed frameworks for creating custom slices of data for evaluation. Many of these have been proposed for natural language processing (NLP) applications, which we also focus on. For example, Errudite proposes a domain-specific language (DSL) for finding and grouping instances based on linguistic features (e.g., the presence of a "person" entity or the number of tokens in the example) [55]. Robustness Gym similarly allows users to construct subpopulations based on linguistic features [18]. Checklist enables generating slices of examples using specific user-defined templates (e.g., `I like {blank}`, where blanks are filled with suggestions from a language model) or transformations (e.g., take an existing set of generated examples and replace proper nouns) [42]. While their goals are related to ours, these systems are designed for developers with technical expertise to identify or generate syntactically-focused groups of examples, and test universally-desirable linguistic capabilities (e.g., "does the

model understand negation?") rather than for end users to specify context-specific behavior on semantically-meaningful slices of data (e.g., "does the model flag comments about diet advice?"). Moreover, Kaleidoscope's generalization process enables discovery, while other systems typically require users to precisely define the examples of interest upfront. We perform a more detailed comparison between the design affordances of template- and DSL-based methods versus Kaleidoscope in Section 4.

3 KALEIDOSCOPE

In this section, we describe the steps of Kaleidoscope's iterative workflow and how we instantiated them in an interactive user interface[1]. To make these sections more concrete, we first introduce a running case study that we utilize throughout.

3.1 Running Case Study: Content Moderation

We use a running case study through the rest of the paper to reason about and instantiate the system with real examples, and illustrate the implications that different contexts can have on model evaluation. We choose a case study for which ML-based tools are currently being developed and deployed to make these analyses more concrete, to allow us to recreate a realistic evaluation by using publicly available models and real-world data, and to enable a user study with participants familiar with the domain [13].

Social media platforms and other online forums are an increasingly common venue for discourse, and often, online harassment. A recent Pew Research Center survey found *"41% of Americans have been personally subjected to harassing behavior online, and an even larger share (66%) has witnessed these behaviors directed at others"* [39]. Recent efforts have tried to use technology to help with comment moderation efforts — for example, by building machine learning models to identify posts or comments that violate rules [1, 5, 22, 31]. These moderation systems can be used in a variety of ways, from helping human moderators prioritize what to look at, to allowing readers to filter which comments they see.

Content moderation is a prime example of a domain in which norms (and consequently, desired model behaviors) differ widely across different contexts. For example, Reddit has over 2 million subreddits, each of which has their own set of rules [7, 15]. Even when rules are shared (e.g., "be civil"), the ways in which they are interpreted can vary (e.g., the comment "Thank you for exposing your Jewishness!" has high inter-rater variability for toxicity [51]). Here, we consider the question of how users or moderators of a particular online community can understand the strengths and limitations of an automated moderation system and assess whether it is suited to their context.

We use Kaleidoscope to look at two publicly available content moderation algorithms: (1) the original Detoxify model released by Unitary (a company that builds moderation tools), trained on Wikipedia comments with crowdsourced toxicity ratings [22]; and, (2) the offensive language identification model released by TweetNLP (an NLP library providing a range of models built with Twitter data), trained on tweets with crowdsourced ratings for offensiveness [5]. We chose the Detoxify model because it is the most highly downloaded comment moderation model (with around 2.08 million downloads) on Huggingface [54], a platform for open source models. We chose the TweetNLP model as a contrast because it is trained on a different data distribution, and we were interested to see if the system could reveal ways in which the two models exhibit different behavior.

Kaleidoscope also requires data from which users build sets of examples. Ideally, this should be data sourced from the target deployment context. In the content moderation example, we consider datasets from different subreddits (i.e., each subreddit is a specific context). We create these datasets with comments that are both unmoderated (i.e., still available on Reddit) and moderated (i.e., had been removed by a moderator). We obtained the unmoderated comments by scraping Reddit with the PushShift API [2], and the moderated comments from a dataset collected in prior work [7]. We subsampled each subreddit dataset to 15,000 examples (10,000 unmoderated and 5,000 moderated).

3.2 Iterative Workflow

Kaleidoscope involves an iterative workflow in which users define meaningful context-relevant concepts and test model behavior on them (Figure 1).

3.2.1 Identification. In the identification stage, users identify a few exemplars of a particular concept they wish to define. Users familiar with the deployment context might draw on prior experience to either create the exemplar(s), or query all examples for a particular word, phrase or regular expression and choose from the results. For example, consider a user who wants to test how well the model moderates xenophobic attacks. They might use a particular comment they have seen as an exemplar, or search for all comments containing the word "immigrant," choosing a few that match their intent. The search process might also be more exploratory—for example, our interactive user interface includes a 2D projection of all comments in the dataset, where users can rapidly mouse over areas or clusters to identify different groups of examples. This stage allows users to employ a bottom-up approach—starting with a small number of concrete examples and then iteratively generalizing to a larger set—rather than a top-down one that requires precisely defining the full slice of interest upfront.

3.2.2 Generalization. In the generalization stage, a few examples are expanded into a larger set of examples that represent the higher-level concept. For example, in the identification stage, a user might identify the single comment "immigrants don't belong here," as violating a norm disallowing xenophobic attacks. In the generalization stage, they would expand this comment into a set of different comments from the dataset that capture the general concept of "xenophobic attacks."

Kaleidoscope enables generalization using iterative content-based retrieval. A user starts with their identified example(s), using them as a seed to search for similar examples. A set of the most similar examples are retrieved using a distance metric in a learned embedding space, and clustered by similarity. Computing distance and retrieving examples in a learned embedding space facilitates finding semantically-related examples (as opposed to generalizing

[1]Our code for both Kaleidoscope's underlying workflow and the UI is available at https://github.com/harinisuresh/test-cases/tree/master.

[2]https://reddit-api.readthedocs.io/

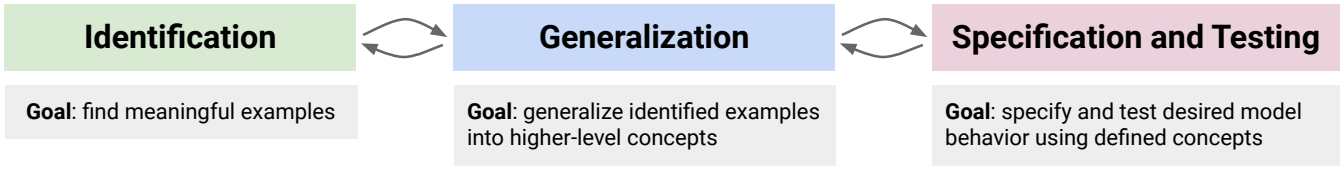

Figure 1: Kaleidoscope's workflow consists of identifying meaningful examples, generalizing them into larger, diverse sets representing important concepts, and using these concepts to specify and test model behavior.

using low-level or syntactic features). Users can then select and add entire clusters or individual examples that fit the desired concept to an *example set*. This process may repeat multiple times, with an expanding set of seed examples.

Facilitating the generalization process is critical, since manually generating large sets of examples is both time-consuming and difficult. While users might be able to identify an example of an important concept, synthesizing many diverse examples representative of the true data distribution is a harder cognitive task [21].

The iterative process also allows users to switch between exploratory and confirmatory analyses. A user might start off with an initial idea of a concept ("xenophobic attacks") and a loosely defined mental model of what this concept encompasses. As they iteratively explore similar examples in the dataset, the bounds of this concept might evolve and become more precisely defined, or the concept might split into multiple (e.g., "anti-Semitic attacks" and "anti-Asian attacks"). They might keep adding similar examples, or step back and cast a more exploratory net by searching all examples for a different word or phrase. Switching between these modes allows users to both discover and instantiate a wide range of concepts.

The steps of the workflow need not be linear; during the generalization process for a particular concept, a user might come across distinct examples that become exemplars for other concepts (e.g., while generalizing "xenophobic attacks," they might come across an example mocking someone for being offended—this might then seed a different "insults about being sensitive" example set).

While we have been illustrating these steps with the content moderation case study, they only require a meaningful representation space in which to compute distance, and some way to visualize the resulting examples. As a result, the workflow can be applied to different application domains or data modalities by selecting a relevant embedding space, and drawing from data visualization techniques from that domain to display examples.

3.2.3 Specification and Testing. In the specification and testing stage, users specify and examine model behavior on the defined concepts. Specifying desired model behavior serves an important role in transparency and trust. Prior work has formalized human-AI trust as *contractual*—i.e., trust is built on an explicit, context-specific contract that specifies the expected behavior of the system [25]. The model behaviors defined in the testing stage can serve as part of such a contract. Importantly, these behaviors are built on top of concepts defined in the generalization phase that align with users' existing mental models of the domain.

We distill two distinct axes used to specify model behavior (see Table 1). The first axis is the behavior type, which describes desired values or shifts in model outputs. For example, Kaleidoscope provides three *behavior types*: 1) specifying the desired model `output`, 2) specifying a desired `invariance` in model outputs, or 3) specifying a desired `directional` change in model outputs. The first behavior type looks at static model outputs, while the latter two behavior types look at shifts in model outputs.

The second axis, *granularity*, applies to behavior shifts, and describes whether the comparison being made is at the `concept-level` or `instance-level`. `Concept-level` shifts consider two example sets, and ask whether there is a statistically significant change in model predictions between them. `Instance-level` shifts ask whether there is a statistically significant pairwise change in predictions after applying a transformation to each example in an example set.

For instance, an `output` test might specify that the model should flag examples in the "xenophobic comments" example set as moderated. A `concept-level invariance` test might specify that model outputs should not be significantly different between an "anti-Asian comments" example set and an "anti-Semitic comments" example set. And an `instance-level invariance` test might specify that model outputs should not change significantly after replacing "Jewish" with "Asian" for each example is the "anti-Semitic comments" example set.

We include both `instance-level` and `concept-level` shifts, since they play different but important roles and entail different tradeoffs. Prior testing frameworks have primarily examined `instance-level` shifts (assessing if model outputs change after applying a transformation to the data) [42, 52, 55]. `Instance-level` tests are useful because they test hypotheses explicitly by constructing counterfactual examples where the input stays constant outside of a defined transformation. However, these tests might produce out-of-distribution or unrealistic examples. For example, offensive anti-Semitic comments likely look different than offensive anti-Asian comments in complex ways, which would not be accurately captured by simply replacing the word "Jewish" with "Asian."

`Concept-level` tests try to account for this by comparing two realistic, independent distributions of data. With `concept-level` shifts, however, it is difficult to precisely attribute the cause of a shift in model behavior. For instance, if in the data used to create example sets, anti-Semitic comments are usually much shorter than anti-Asian comments, and we find that the model is more likely to flag them as moderated, it is unclear whether this is because of their content or their length. Findings from `concept-level` tests

Behavior Type	Granularity	Definition	Example in words
Output	—	$mean(\{\mathbb{I}[f(A_i) = \hat{y}]\}_{i=0}^{\lvert A\rvert})$	The model should predict that *xenophobic attacks* (A) should be moderated.
Invariance	Concept-level	$mean(\{f(A_i)\}_{i=0}^{\lvert A\rvert}) - mean(\{f(B_i)\}_{i=0}^{\lvert B\rvert}) < e$	The model's predictions should not significantly differ between *xenophobic attacks* (A) and *sexist attacks* (B) .
	Instance-level	$mean(\{f(A_i) - f(t(A_i))\}_{i=0}^{\lvert A\rvert}) < e$	The model's predictions should not change significantly after adding "lol" to each example in *xenophobic attacks* (A) .
Directionality	Concept-level	$mean(\{f(A_i)\}_{i=0}^{\lvert A\rvert}) - mean(\{f(B_i)\}_{i=0}^{\lvert B\rvert}) > e * d$	The model should predict that *xenophobic attacks* (A) are more likely to be moderated than *civil discussion* (B) .
	Instance-level	$mean(\{f(A_i) - f(t(A_i))\}_{i=0}^{\lvert A\rvert}) < e * d$	The model's predicted probability of moderation should increase after replacing "you" with "you prick" in *civil discussion* (A) .

Table 1: Model Behavior Specification. Output tests check whether the predictions of a model f on example set A align with a desired output $\hat{y}$ (Row 1). Concept-level tests compare two example sets A and B, and check whether the distribution of model predictions significantly differs between the two (Rows 2 and 4). Instance-level tests instead compare example set A, and a user-specified transformation t of A, which is applied to each member of the input example set (Rows 3 and 5). Directionality tests also involve a specified direction $d \in \{-1, 1\}$, indicating whether the difference should be positive or negative. Both invariance tests and directionality tests are governed by a threshold e at which the distributions of model predictions may be deemed significantly different, and can be determined by a statistical test (e.g., a t-test). We also require that the p-value of the statistical test is less than a set threshold.

can still be valuable if the data used to create example sets reflects the actual data distribution and correlations in a particular context, since they provide a lens into the correlations the model would exploit in deployment.

Kaleidoscope provides a selection of model behaviors (e.g., outputs, invariances, directional changes) and transformations for testing `instance-level` shifts (e.g., replacing/adding/deleting words). At the same time, by identifying these higher-level axes and how they fit together, the system is flexible to adding many different types of behaviors and/or transformation functions as they are developed.

3.3 Interactive User Interface

We implement an interactive user interface to facilitate Kaleidoscope's workflow and make the system approachable for who might not have programming experience.

The interface consists of three main panes (see "Overall View" in Figure 2): identification and generalization primarily happens in the leftmost pane, where users can explore and find examples (B) to add to new or existing example sets (A). Specification and testing happens on the rightmost pane (D), where tests are displayed. The middle pane (C) contains a 2D projection plot where examples can be moused over or selected. Color and shape encodings highlight examples and/or model predictions when an example set or test is expanded.

As an illustrative example, we walk through creating an example set representing criticisms or abuse directed at moderators (a concept that is typically moderated across many different subreddits) using data from r/news. Screenshots from this process are displayed in Figure 2, and in the following sections, numbers in parentheses reference specific screenshots.

3.3.1 Identification. To start, a user could either write a seed example (e.g., based on prior experience) or find one in the dataset. To find one, they can use the search bar to search all examples for the word "mods," which they imagine will appear in many relevant examples. In the Search and Explore section, the system returns all comments containing the word "mods" clustered into three groups by similarity **(1)**.

To cluster examples, the system uses K-means clustering in a learned embedding space. The embeddings are computed by the Universal Sentence Encoder [6], a publicly-available transformer-based language model. The points in each cluster are highlighted in the projection plot, and the top words in each appear below (**1a**).

Displaying the result in clusters helps users parse high-level structure in the returned examples. The top words provide an additional summarization of each cluster to help with this sensemaking. Skimming through the examples and top words can help a user get a sense for the types of examples in each cluster: the first with longer rants or discussions about mods, the second with anti-Islamic insults against mods, and the third with more general short, hostile statements against mods. A user's domain knowledge could guide which examples to select to seed an example set, or whether the examples returned comprise distinct enough types that we might actually want to create multiple example sets (e.g., anti-Islamic criticism as well as general criticism).

In our example, a user might determine that examples from the latter two clusters should be treated similarly. They can select the examples from both to seed a new example set named "insulting

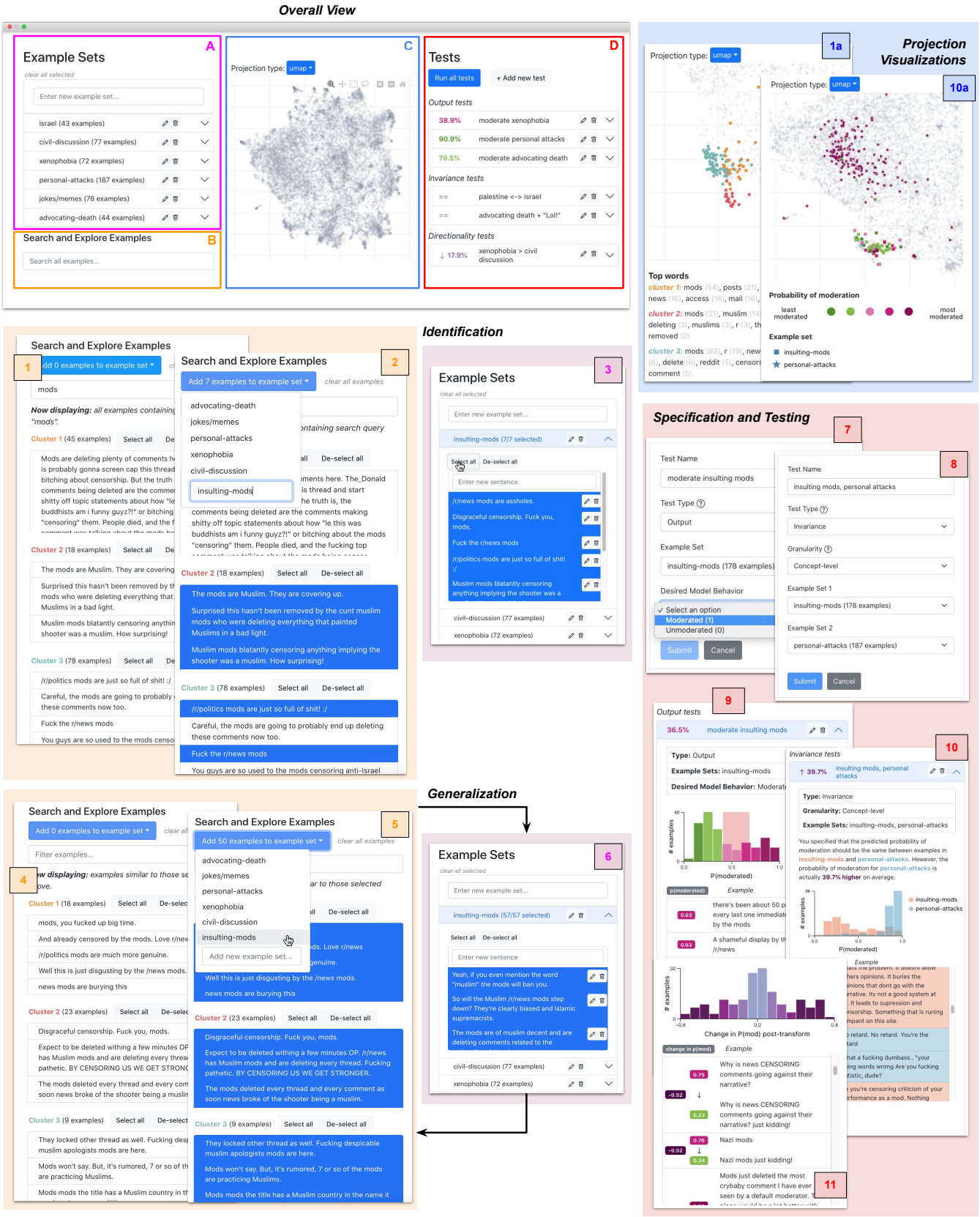

Figure 2: Kaleidoscope's interactive user interface, and how different parts of the iterative workflow happen within it. See the text for a step-by-step walk-through.

mods" **(2)**. This example set now appears in the list of example sets, and contains the set of examples they chose.

3.3.2 Generalization. They can now click to select all (or a subset of) these examples **(3)**. This creates the *selected set*, or examples used to retrieve other, semantically similar examples from the data. These similar examples are populated in the Search and Explore section **(4)**.

The system finds similar examples by using Euclidean distance with embeddings from the Universal Sentence Encoder. It computes the mean embedding of all examples in the selected set, and finds the most similar examples to that mean vector. The returned similar examples are also clustered, and a user can follow a similar process as before—getting a high-level sense of each cluster, and then choosing to either add entire clusters, or specific examples, to the example set.

As they add more examples to the "insulting mods" example set **(5)**, these examples are automatically added to the selected set **(6)**. The similar examples continue to update to display ones similar to all of the now-selected examples (repeating **4**, **5** and **6**). Searching by semantic similarity reveals examples that do not necessarily contain the specific search terms we might have thought about. For example, returned similar examples include "I just came from /r/undelete and holy hell is the censorship here is bad," and "So why the fuck would you delete the mass upvoted post that was originally posted?" These are comments implicitly about or addressed to moderators, but which they would not have found with a string or regular expression match.

As they iteratively find and add examples, a user's mental model of the example set and concept evolves and becomes more precise. Initially, the user may have had the broad idea of capturing insults against mods. Looking at real examples (both individual and higher-level clusters), lends clarity to bounds of this concept (e.g., they chose to focus on shorter, aggressive insults, as opposed to longer discussions) and what it includes (e.g., specific attacks about censorship, anti-Islamic rhetoric, comparisons to other subreddits).

They can also switch between exploration and confirmation. Finding and adding similar examples utilizes a particular type of example the user has confirmed is relevant. During this process, however, they might see a phrase or word that spurs them to zoom out and return to an exploration stage, searching all examples for a different phrase to see how else it appears in the data. For instance, they might notice the word "censorship" appearing in some of the returned examples, and use that as a search query to search all examples, casting a broader net before drilling down again. Similarly, they could use the projection plot to select the broader area around where points in the example set are located. This populates the Search and Explore section with those examples, and allows them to find examples they might have missed.

As the user continues to iteratively add similar examples, they will typically observe one of two behaviors: 1) *convergence*, where similar examples become increasingly similar, until they are not significantly different from the ones already added, and no longer diversify the example set, or 2) *divergence*, where examples being returned are not actually similar in relevant ways. Which behavior they observe is highly dependent on how well-represented a particular concept is in the dataset (i.e., if there were very few examples of insults against mods, the retrieved similar examples would quickly become divergent). This is also one of the limitations of our method; while examples are realistic because they are drawn from real data, we also inherit the limitations of that data. We examine this limitation further in the discussion.

In the "insulting mods" example, the user might begin to observe a convergence after 4-5 iterations of adding similar examples, where new retrieved examples seem repetitive, rather than adding additional diversity. At this point, they have 187 examples, and can choose to create a test using this example set.

3.3.3 Specification and Testing. When a user clicks "Add new test," a form appears to specify different axes of desired model behavior. Depending on the behavior type they select (e.g., output, invariance, etc), different parameter options are provided **(7, 8)**.

To test how well the model moderates comments insulting moderators, they can create an output test, specify the desired behavior as moderated, and run the test. In their header, output tests display the percentage of the examples that have the desired behavior—in this case, the user can see that the model only predicts that 34.2% of these comments should be moderated. The system also provides a more detailed output when the test pane is expanded **(9)**. This allows users to explore the distribution of predicted probabilities (with a histogram visualization), as well as outputs for individual examples (with an output log). Brushing over the histogram of predicted probabilities filters the examples, so a user can examine, for instance, specifically the ones that were moderated with high probability. Viewing individual examples makes this analysis concrete, allowing users to investigate, for instance, whether the examples with a higher moderation probability actually are more severe violations than the ones with lower probabilities.

In their headers, `concept-level` shift tests display the mean difference in predicted probability across the two example sets in the test, and `instance-level` shift tests display the mean pairwise difference across an example set pre- and post-transformation. Displaying the actual difference (rather than just pass or fail, for example) provides richer signal into how well or poorly the model does, and helps compare results across tests.

When expanded, `concept-level` tests display a probability histogram and example log containing both example sets in the test encoded via different colors **(10)**. This allows users to compare the overall distributions of predicted probabilities, as well as compare examples from each example set that fall within a given probability window. The expanded view of `instance-level` tests shows a probability histogram of the pairwise differences between the predicted probability of each example pre- and post-transformation. The example log displays each example, its predicted probability before and after the transformation, and the difference in those probabilities **(11)**.

When a test pane is expanded, the projection plot displays the points in the example set(s) being tested. Color encodes predicted probability, and shape encodes example set membership for concept-level tests **(10a)**. This visualization allows users to see if there are high-level patters in the model's predictions (e.g., certain clusters that are highly moderated or not) and drill down into the plot to characterize them.

4 EVALUATION: COMPARING CONCEPTUAL AFFORDANCES

In this section, we compare Kaleidoscope to other ML model evaluation systems that group examples via template- or DSL-based methods. We focus on Checklist [42] and Errudite [55], respectively, as examples of these alternate classes of evaluation frameworks. To guide our analysis, we draw on the Cognitive Dimensions of Notation framework [19], which includes several axes of comparison (e.g., *hidden dependencies*, *premature commitment*) for systems such as programming languages or visual interfaces.

Using Checklist, users specify a template (e.g., `I really {mask} the flight`) and can fill in the blanks with suggestions from a predefined lexicon or a language model (LM); or, they can apply perturbations to examples from an existing dataset. With Errudite, users write filters using a DSL to query an existing dataset based on linguistic features (e.g., the filter `count(token(x, pattern="PERSON")) > 2` would return examples where there are more than two `PERSON` entities).

We find that Checklist, Erudite and Kaleidoscope have significant differences in the amount of *premature commitment* and the type of *hard mental operations* required. They also involve different *abstractions* and *hidden dependencies*.

Consider the example of making a test for a content moderation system, to check whether "political comments" are moderated (a subreddit rule in r/funny). We first used Checklist to try and test this behavior. We started by manually defining a custom lexicon of `political_figures` ("Hillary Clinton," "Mitch McConnell," "The President," etc.). We used LM suggestions to fill in the template `{political_figure} is a {mask}`, saving those suggestions into a `descriptive_noun` lexicon. We used LM suggestions again to fill in the template `political_figure is a {mask} {descriptive_noun}`, saving them into an adjective lexicon. We then used Checklist to generate all combinations of these templates and specified that their label should be moderated, resulting in 30,600 generated examples. We went through a similar process to generate another set of examples of the form `{political_group} has {adjective} views on {political_issue}` in addition to `{political_group} is {descriptive_noun}` where we manually defined sets of words for `political_group` and `political_issue`. This resulted in 8800 generated examples.

Errudite's DSL involves more formal *linguistic abstractions* than Checklist's templates—e.g., specific entity types or part of speech tags. To try find "political comments" with Errudite, we manually created a list of tokens representing political figures, as in the prior example, and wrote a filter to find examples containing those entities (e.g., `has_any(token(x, pattern="PERSON")), ["The President", ...]`). This resulted in 124 examples. Some of these examples did not necessarily belong in "political comments" (e.g., "Donald Trump had a cameo? I must have missed it.") but were returned because they contained a matching entity, and there was not a different attribute with which to filter them out. The DSL allows users to compose filters into increasingly complex queries (e.g., constraints on the types of entities, comment lengths, etc), but these are not as useful for our use case, where examples of interest do not group by these linguistic features, but rather, by higher-level semantic features that are difficult to precisely formalize.

With Kaleidoscope, we started by searching for a term we expect to appear in lots of political comments (e.g., "Trump"). The results were grouped into three clusters, each using the word "Trump" in different contexts: the first containing longer discussions about race, the second about the election, and the third with short, aggressive insults. We expect knowledge about the context to guide the breadth/granularity of the example set. Are aggressive insults about political figures distinctly worse than longer discussions, or would all of these political comments be moderated regardless of tone? This could inform whether to seed one or multiple example sets. Here, we chose a particular set of examples to seed an example set, and then generalized them into a larger set by searching for similar examples. Looking at semantically similar examples revealed other types of relevant examples that we did not think of from scratch — for example, comments that contained semantically-related phrases or people, such as "make america great again," "gun-owning republican," or "Bernie." The resulting example set contained 272 examples. We also ended up seeding new example sets based on related but distinct types of comments that appeared during generalization (e.g., sets on "detailed political issue discussions" and "insults about being triggered").

For each system, this process of creating sets of examples requires a different set of *hard mental operations* and *hidden dependencies*. With Checklist, we needed to keep track of different sets of words (`political_figures`, `descriptive_nouns`, etc.) and compose them into templates that made sense. It was difficult to generate diverse template formats, and to assess whether we had specified enough of them. While templates are already more abstract than actual examples, Errudite's DSL involves an additional layer of *abstraction*. Because of this, trying to keep track of the mapping from a particular DSL-based filter to the concrete examples specified by it involves several mental jumps. Kaleidoscope does not use linguistic abstractions to define example sets—rather, the example set is simply defined by the concrete examples it contains. Examples are familiar and intuitive to users, and working with them is straightforward. However, templates and DSLs do create a formal definition of the example set; i.e., the dependency between a template or filter and the contents of the resulting example set is *explicit*. In Kaleidoscope, because the example set is less precisely defined, we had to keep track of the types of examples that were included, and continue updating this mental model as we iteratively added examples.

Checklist and Errudite also require significantly more *premature commitment* than Kaleidoscope. For example, with Checklist, we were able to generate thousands of examples, but they follow a very specific template which we were required to formulate at the beginning. Kaleidoscope's process is more bottom-up—we started with a general idea of a word that would be present in political comments, and seeing the distribution of real examples from this context helped clarify the bounds of our hypothesis. We end up with fewer examples; however, they are more varied, and a more accurate reflection of how political comments actually look in context.

As a result of these differences, the tests we created also tell us different things. With Checklist, we had high confidence in the model's behavior on examples following the specific templates we wrote, but were uncertain about how this might generalize to the natural data distribution. With Errudite, examples are drawn

from the real dataset, but the only filter that applied to our use case was a coarse string match that resulted in a skewed sample—returning some irrelevant examples that contain query tokens, and missing a lot of relevant examples that do not. We found that the Checklist tests had 99% and 86% failure rates, and Errudite's test had an 80% failure rate (i.e., saying that most political comments go unmoderated). With Kaleidoscope, our test on political comments had a 66% failure rate. The difference in these results indicates that the model is more likely to moderate political comments from the real data distribution. For example, this might reflect the fact that in this context (the r/funny subreddit), comments about politics are more likely to be aggressive or insulting. If our goal is context-specific evaluation, Kaleidoscope allows us to understand the type of behavior we should anticipate in this particular context.

The design affordances of each tool make them suited to different types of analyses. With Checklist and Errudite, it is easier to test a range of general linguistic capabilities (e.g., if the model is robust to replacing neutral words with other neutral words, or if it can deal with sentences that have complex linguistic structures). On the other hand, Kaleidoscope is better suited to creating topic-oriented example sets ("aggressive comments", "political comments") and tests that reflect semantically-meaningful goals.

5 EVALUATION: USER STUDY

To understand how users might go through Kaleidoscope's workflow and interface to assess a model's suitability for their contexts, we conducted a study with 10 users and 3 moderators from Reddit. The study was certified by our institution as exempt from full IRB approval under category 3 (benign behavioral intervention).

5.1 Study Methods

We recruited our participants by posting to r/SampleSize (a subreddit for posting studies), messaging individual moderators on Reddit, and emailing our institutional networks. We filtered participants to those who reported spending 5+ hours on Reddit per week. For each study, we seeded the system with data from a subreddit that the participant was familiar with, and two publicly available moderation models. The system uses the original Detoxify model [22] by default, but we told participants that they could also switch to compare a second model (TweetNLP's offensive language classifier [5]) via the settings tab if they wanted. See Section 3.1 for more details on these datasets and models.

Each study lasted between 48 and 62 minutes and participants were paid $20. We spent 15 minutes introducing the project and demonstrating the interface by creating an example set of insults against moderators, and testing that it should be moderated (similar to the example in Section 3.3). We then asked participants to imagine that their subreddit was considering adopting an automated moderation model, and that their goal was to use Kaleidoscope to better understand the strengths and weaknesses of this model and assess if it would be suited to their context. As an initial prompt, we asked them to think about types of comments that came to mind that would be concerning or in violation of subreddit rules, and that they would want to make sure an automated moderation system would know how to deal with. Rather than ask everyone to complete the same tasks, we took this approach to evaluate Kaleidoscope's effectiveness for *context-specific* analysis.

For the majority of the remaining study time, users then continued to use the system independently to create, modify, and explore example sets and tests. As facilitators, we answered simple mechanistic questions about the system and user interface but did not provide further instructions as to what they should test. We ended with a short debrief where we asked how participants felt about the model being used in their context, based on what they had learned about it using Kaleidoscope—e.g., whether they thought the model was well-suited or not, and how they thought it compared to other forms of moderation currently being used.

To analyze the studies, we rewatched all video recordings and extracted quotes or actions that related to how participants approached creating example sets and tests, and/or how they reasoned about or reflected on the model. We iteratively annotated and grouped these into themes — starting with a few a priori hypotheses about expected user behavior (e.g., that they would discover relevant new search terms in the retrieved similar examples), but iterating on and modifying them as we reviewed the data (i.e., a combined inductive and deductive approach [2]). We highlight prominent themes in Sections 5.3 - 5.6.

5.2 Overall usage

Participants created example sets spanning a broad range of topics, based on their personal experience (e.g., offensive examples they had encountered, or posts of theirs that had been moderated) or specific subreddit rules. Examples sets that participants created included "colorism," "self-promotion," "personal attacks," "covert racism," "LGBT attacks," "sexism," "civil discussion about race," and "piracy/torrenting." When specifying tests, participants primarily created output tests (four participants also created tests about behavior shifts).

It took users between 3 and 12 minutes to create an example set, with an average size of 122 examples. When going through the generalization process of building out an example set, participants typically added groups of examples (e.g., an entire cluster of similar examples). They were able to skim the examples and top words to get a high-level sense of an entire cluster, rather than verifying the relevance of each example individually.

5.3 Iterative generalization enables discovery

To create example sets, participants typically started with a search query of a term they expected to appear in relevant examples—for example, using the search query "gay" to find examples to seed an "LGBT attacks" example set. They would then perform several rounds of finding and adding similar examples, until they found that new similar examples were either out-of-scope or repetitive. At this point, they often stepped back and diversified by trying a different but related search query (e.g., searching "trans" for "LGBT attacks"), picking some examples, and repeating the generalization process with these new examples as seeds.

They often discovered these additional search queries through noticing words or phrases in similar examples, and realizing that they might reveal a different subset of the concept at hand. For

example, one participant, creating an "attacks against liberals" example set, initially searched for "liberal," and started generalizing based off of some selected examples. One of the returned examples contained the term "SJW," which she recognized as a pertinent term that might appear in a range of other relevant examples, and used it as her next search term. It also prompted other subsequent, related, search terms (e.g., "snowflake"). Other participants discovered different spellings of words (e.g., "p1rate bay") or acronyms they hadn't thought of (e.g., "BLM") that they used as additional search terms. Through this iterative discovery process, participants' mental models of the concepts evolved and expanded, covering additional types of examples they had not initially thought about.

5.4 Iterative generalization helps draw out implicit knowledge

The generalization process lent clarity to the bounds and contents of the concept in other ways as well—for example, as users delineated which similar examples did or did not belong in the concept, either at the individual example or cluster level. Several participants noticed implicit subgroups within similar examples and split their initial ideal into multiple concepts. For example, one participant initially intended to create a "racism" example set. While looking at retrieved similar examples, she realized that there was a distinction between comments that were outrightly offensive and those that disguised racist sentiments behind lengthy arguments. She ended up creating two example sets representing these two subsets of examples. Another participant started to create a "self-promotion" example set, but noticed several comments returned in the similar examples that fit into what she called "general spam" rather than specifically self-promotion. She ended up creating an additional "general spam" example set seeded with those examples. These are distinctions that the participants might have had difficulty identifying upfront — but the generalization process helped draw out this implicit knowledge. This might be because viewing data from their context makes reasoning about distinct types of examples familiar and intuitive.

5.5 Output tests help reason about context-specific tradeoffs

Running and exploring the results of `output` tests helped participants reason about if and how they would use the model in their context. In particular, because tests operate on semantically-meaningful concepts, participants were able to contextualize model behavior in relation to existing moderation methods. For example, a moderator of r/TIFU created an example set representing "fake callouts" (claiming that others' posts/stories are fabricated—these comments are typically removed in that subreddit). They created an output test to specify that these posts should be moderated, and found that the model's performance was 63% (of the 85 examples in the example set, it predicted 65% should be moderated). While compared to typical ML standards, this performance is quite poor, the participant was excited by it: "this could be helpful [...] If it's flagging that much, you know, that's outperforming all the moderators out there and catching stuff they wouldn't." Another participant responded in a similar vein to the model having 80% accuracy on an example set of disrespectful comments: "It's not as accurate as I'd hoped, but I'd still use this model. It's incredibly hard to moderate on my own, and this could be useful, especially if it was used in tandem with a human moderator to filter posts."

In another case, a participant reasoned about suitability across different contexts. He created a "piracy" example set, which he felt were an important type of comment in r/movies that an automated moderation system would need to deal with. However, he found that the model only moderated 8% of examples in that example set. He also created an example set on "personal attacks," and found that the model had 98% performance on it. He subsequently expressed doubt that the model would be suitable for r/movies, but suggested that it might be helpful for another subreddit he moderated, where the bulk of comments that are moderated "are more daft arguments, blatant insults."

Participants also found the more detailed reports from each test useful for understanding model behavior beyond the percentage correctly predicted. In several cases, the model appeared to have subpar performance on a particular output test, and examining the log of individual comments and predictions lent clarity into whether that performance was acceptable or not. For example, in some cases, when participants created output tests specifying that an example set should be moderated, they would look at the specific examples that were not moderated by the model, and find that they were less severe than the other examples—e.g., "These aren't really the worst thing – most of the really bad ones were caught so that's actually useful." Participants felt that an automated moderation system should be used in tandem with a human moderator, so if it was erring on the side of moderating less (and catching the most severe violations), they felt satisfied with its performance. In other cases, examining individual examples and predictions made participants less confident in the model — for example, if the model's decision boundary seemed random, did not agree with participants' prior expectations, or appeared to be reliant on unimportant features.

5.6 Testing behavior shifts reveals important model weaknesses

Participants who tested shift behaviors also discovered interesting strengths and limitations about the models that impacted their confidence. One participant, for example, created an example set of "white supremacist dog whistles," and found that adding "thanks for reading" to the end of each comment (via an `instance-level invariance` test) decreased the probability of moderation by 21% for the Detoxify model. The probability of moderation stayed the same using TweetNLP's model, which provided useful insight: "I'd want to look into the second model further, since the first is pretty problematic." Another created an example set of random, benign comments, and found that adding "Yes, I'm gay" to the end of each (also via an `instance-level invariance` test) increased the probability of moderation by 26.2% for the Detoxify model and 52.4% for TweetNLP's model. Together, these tests show these models entail different weaknesses that our system can help characterize.

Others used `concept-level` and `instance-level` shifts together to reveal different things about the model. For example, one participant created example sets representing "homophobic attacks" and "transphobic attacks". They created a `concept-level invariance` test to specify that these two example sets should be treated the

same, as well as an `instance-level invariance` test with "homophobic attacks," where they applied a transformation replacing the word "gay" with "trans" in each comment. The `concept-level` test revealed that "transphobic comments" were 25.3% less likely to be moderated than "homophobic comments," while the `instance-level` test reported that predictions were not significantly different after applying the transformation. This difference highlights that the way that "homophobic attacks" and "transphobic attacks" manifest in this context is different, and that simply replacing the word "gay" with "trans" (while the rest of the comment stays the same) does not fully capture that difference. While the participant considered robustness to switching the attack target (demonstrated by the `instance-level` test) a desirable behavior, they held reservations about the model's performance if deployed, given the subpar performance on the `concept-level` test (which better reflects the real-world distribution of comments).

5.7 Limitations

Participants found certain aspects of the system confusing. A common confusion occurred when they observed divergent behavior during generalization, typically due to trying to create a particular example set that was not well represented in the data. For example, one participant tried to create an example set on "positive LGBT discussions," using a dataset from r/funny, but found that the similar examples kept diverging towards negative comments, which were much more present in that context. Others were interested in specific topics (e.g., "China" or "celebrity news") that were not well represented in the data, and thus difficult to represent in example sets. Several participants also brought up that it was difficult to evaluate certain comments without the surrounding context (i.e., what they were written in response to). We chose not to include this context to mimic the way that the models (which do not take context into account) would see examples; but in doing so, participants' experience using the tool felt inconsistent with how they would typically encounter examples.

In addition, the current study design has limitations. We conducted a qualitative observational study so that we could observe participants use and think aloud about the system in real-time. We did not attempt to measure quantitative metrics of trust or behavior change as we believe that these metrics will only reflect meaningful signal after sustained, engaged use with the system. Finally, while Kaleidoscope's underlying workflow is applicable to different domains and data modalities, our evaluation only focuses on the content moderation case study. Additional studies are needed to understand if our observations generalize to other user groups and application domains.

6 DISCUSSION AND FUTURE WORK

We present Kaleidoscope, an iterative workflow and interactive user interface for user-driven, context-specific model evaluation. Rather than use static tests sets or pre-defined data slices, Kaleidoscope presents an alternative paradigm for model evaluation that allows on-the-ground users to identify examples of important concepts, generalize them into larger, representative sets, and specify and test model behavior with them in semantically-meaningful ways.

Through a comparative evaluation using the Cognitive Dimensions of Notation framework [19], we show how other methods to group examples ask users to define formal definitions of data slices, which requires significant premature commitment and linguistic expertise. Kaleidoscope's generalization process instead enables discovery, and is grounded in real examples. In a study with reddit users/moderators, participants found the interactive process of finding and adding similar examples intuitive, and created a range of example sets populated with diverse examples that would be difficult or impossible to specify via a template or DSL. The resulting example sets reflect semantically-meaningful, context-relevant concepts (e.g., "covert racism" or "LGBT attacks"). Kaleidoscope enables specifying and testing a range of model behaviors using these concepts. Specifying tests makes desired behavior transparent, and running them reveals relevant insights into model strengths and weaknesses that help users reason about how the model would perform in their context.

We note some of the current limitations of our system, and their implications for future directions. Kaleidoscope trades off precision for flexibility, allowing users to create example sets that are so varied it would be extremely difficult to define them via formal linguistic abstractions (e.g., template or DSL). In doing so, however, it also requires users to keep track of the types of examples they are adding and update their mental model of the example set. Users can assess coverage by observing whether retrieved similar examples are continuing to add diversity to an example set, but this is a heuristic measure (not a guarantee, as in Errudite [55], for example).

We imagine two broad directions for addressing this issue in future work: the first focused on making it easier for users to assess the contents and boundaries of example sets, and the second more computational, focused on methods that facilitate creating example sets with higher coverage. The first direction could draw inspiration from data summarization and visualization [17, 29, 47, 53]—for instance, highlighting distinct exemplars in the set, or visualizing existing or learned features of the examples beyond top words (e.g., length, sentiment, tone). The second direction could explore extensions to our example retrieval method—for instance, rather than finding and returning the most similar examples, we could find and return similar examples that are also different enough from any example already in the set (e.g., drawing from metrics in coverage-based fuzzing [36]), to encourage creating example sets with diverse examples.

Because Kaleidoscope's example sets are grounded in real data, they also inherit the limitations of the dataset used to seed they system. We intend the system to be used to evaluate a model for a particular context, and the dataset used to be from that context. This helps ensure that users are familiar with the data they see, and that important concepts in that context are likely to appear in the dataset. However, as we found in our user study, sometimes users may want to create example sets that are more hypothetical or less well represented in the natural data distribution. In these cases, similar examples tend to diverge quickly to examples that are not actually relevant to the concept at hand. One possibility to address this issue could be to draw data from other distributions, if we observe that the most similar examples retrieved in the original dataset are further than a specified threshold. For example, a participant in

our user study had trouble creating an example set representing "positive LGBT discussions" in the context of data from r/funny, where negative LGBT attacks are much more common. In this case, Kaleidoscope could potentially draw data from a different source where these examples would be more common (e.g., r/LGBT).

The current work also opens up promising future ideas for model development and participatory benchmark creation. We were encouraged by the wide range of different topics, often drawing from their personal experience, that participants in our user study examined. In the future, we imagine Kaleidoscope could be used to facilitate calls for participatory or crowdsourced benchmarks [10]. Kaleidoscope is well-suited to address this need because the system is not *only* exploratory—it allows users to define example sets representing higher-level concepts and specify expected model behavior on them. For example, individuals or groups could specify what kinds of examples they think fit into a particular concept (e.g., "sexist comments") and how they would expect those examples to be treated by a model. These tests could be compiled and used similarly to a benchmark, for evaluating models and their future iterations. This approach acknowledges that "ground truth" is often subjective and dependent on users' contexts and lived experiences [11, 12, 23, 48, 50], and could help make transparent which people or populations are and are not served by a particular model. Benchmarking methods and datasets drive research agendas and values in ML [10, 14], so this shift has broader implications. Making these processes more participatory shapes future iterations of models and what is considered state-of-the-art, pushing them to prioritize domain knowledge and contextual values [27].

Kaleidoscope contributes to a growing body of work improving human-AI trust and giving users the agency to question, probe, and push back on automated systems. We ask how to address these issues in a way that is fundamentally grounded in context, and our results suggest rich directions for future work in model evaluation.

ACKNOWLEDGMENTS

This research was sponsored by NSF Award #1900991, and by the United States Air Force Research Laboratory under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*. 491–500.

[2] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[3] Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*. PMLR, 77–91.

[4] Carrie J Cai, Samantha Winter, David Steiner, Lauren Wilcox, and Michael Terry. 2019. " Hello AI": uncovering the onboarding needs of medical practitioners for human-AI collaborative decision-making. *Proceedings of the ACM on Human-computer Interaction* 3, CSCW (2019), 1–24.

[5] Jose Camacho-Collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa-Anke, Fangyu Liu, Eugenio Martínez-Cámara, et al. 2022. TweetNLP: Cutting-Edge Natural Language Processing for Social Media. *arXiv preprint arXiv:2206.14774* (2022).

[6] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*. 169–174.

[7] Eshwar Chandrasekharan, Mattia Samory, Shagun Jhaver, Hunter Charvat, Amy Bruckman, Cliff Lampe, Jacob Eisenstein, and Eric Gilbert. 2018. The Internet's hidden rules: An empirical study of Reddit norm violations at micro, meso, and macro scales. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–25.

[8] Anamaria Crisan, Margaret Drouhard, Jesse Vig, and Nazneen Rajani. 2022. Interactive Model Cards: A Human-Centered Approach to Model Documentation. In *2022 ACM Conference on Fairness, Accountability, and Transparency* (Seoul, Republic of Korea) *(FAccT '22)*. Association for Computing Machinery, New York, NY, USA, 427–439. https://doi.org/10.1145/3531146.3533108

[9] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. 2020. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395* (2020).

[10] Emily Denton, Alex Hanna, Razvan Amironesei, Andrew Smart, Hilary Nicole, and Morgan Klaus Scheuerman. 2020. Bringing the people back in: Contesting benchmark machine learning datasets. *arXiv preprint arXiv:2007.07399* (2020).

[11] Mark Díaz, Ian Kivlichan, Rachel Rosen, Dylan Baker, Razvan Amironesei, Vinodkumar Prabhakaran, and Emily Denton. 2022. CrowdWorkSheets: Accounting for Individual and Collective Identities Underlying Crowdsourced Dataset Annotation. In *2022 ACM Conference on Fairness, Accountability, and Transparency* (Seoul, Republic of Korea) *(FAccT '22)*. Association for Computing Machinery, New York, NY, USA, 2342–2351. https://doi.org/10.1145/3531146.3534647

[12] Catherine D'ignazio and Lauren F Klein. 2020. *Data feminism*. MIT press.

[13] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).

[14] Ravit Dotan and Smitha Milli. 2020. Value-laden disciplinary shifts in machine learning. FAT*'20: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (Jan. 2020).

[15] Casey Fiesler, Joshua McCann, Kyle Frye, Jed R Brubaker, et al. 2018. Reddit rules! characterizing an ecosystem of governance. In *Twelfth International AAAI Conference on Web and Social Media*.

[16] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence* 2, 11 (2020), 665–673.

[17] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D Hansen, and Jonathan C Roberts. 2011. Visual comparison for information visualization. *Information Visualization* 10, 4 (2011), 289–309.

[18] Karan Goel, Nazneen Rajani, Jesse Vig, Samson Tan, Jason Wu, Stephan Zheng, Caiming Xiong, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the nlp evaluation landscape. *arXiv preprint arXiv:2101.04840* (2021).

[19] Thomas RG Green. 1989. Cognitive dimensions of notations. *People and computers V* (1989), 443–460.

[20] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

[21] Frank Haist, Arthur P Shimamura, and Larry R Squire. 1992. On the relationship between recall and recognition memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 18, 4 (1992), 691.

[22] Laura Hanu and Unitary team. 2020. Detoxify. Github. https://github.com/unitaryai/detoxify.

[23] Donna Haraway. 2020. Situated knowledges: The science question in feminism and the privilege of partial perspective. In *Feminist theory reader*. Routledge, 303–310.

[24] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HJz6tiCqYm

[25] Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 2021. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 624–635.

[26] Kimmo Karkkainen and Jungseock Joo. 2021. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1548–1558.

[27] Daniel N Kluttz, Nitin Kohli, and Deirdre K Mulligan. 2022. Shaping our tools: Contestability as a means to promote responsible algorithmic decision making in the professions. In *Ethics of Data and Analytics*. Auerbach Publications, 420–428.

[28] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas

Phillips, Irena Gao, et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*. PMLR, 5637–5664.

[29] Kostiantyn Kucher and Andreas Kerren. 2015. Text visualization techniques: Taxonomy, visual survey, and community insights. In *2015 IEEE Pacific visualization symposium (pacificVis)*. IEEE, 117–121.

[30] Deepak Kumar, Patrick Gage Kelley, Sunny Consolvo, Joshua Mason, Elie Bursztein, Zakir Durumeric, Kurt Thomas, and Michael Bailey. 2021. Designing toxic content classification for a diversity of perspectives. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. 299–318.

[31] Alyssa Lees, Vinh Q Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. 2022. A new generation of perspective api: Efficient multilingual character-level transformers. *arXiv preprint arXiv:2202.11176* (2022).

[32] Yunhui Long, Vincent Bindschaedler, and Carl A Gunter. 2017. Towards measuring membership privacy. *arXiv preprint arXiv:1712.09136* (2017).

[33] Donald Martin Jr, Vinodkumar Prabhakaran, Jill Kuhlberg, Andrew Smart, and William S Isaac. 2020. Extending the machine learning abstraction boundary: A Complex systems approach to incorporate societal context. *arXiv preprint arXiv:2006.09663* (2020).

[34] George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 63, 2 (1956), 81.

[35] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*. 220–229.

[36] Augustus Odena, Catherine Olsson, David Andersen, and Ian Goodfellow. 2019. Tensorfuzz: Debugging neural networks with coverage-guided fuzzing. In *International Conference on Machine Learning*. PMLR, 4901–4911.

[37] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems* 32 (2019).

[38] Desmond U Patton, William R Frey, Kyle A McGregor, Fei-Tzin Lee, Kathleen McKeown, and Emanuel Moss. 2020. Contextual analysis of social media: The promise and challenge of eliciting context in social media posts with natural language processing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 337–342.

[39] Pew Research Center. 2017. Online Harassment 2017. https://www.pewresearch.org/internet/2017/07/11/online-harassment-2017/

[40] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2008. *Dataset shift in machine learning*. Mit Press.

[41] Deborah Raji, Emily Denton, Emily M. Bender, Alex Hanna, and Amandalynne Paullada. 2021. AI and the Everything in the Whole Wide World Benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung (Eds.), Vol. 1. https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/084b6fbb10729ed4da8c3d3f5a3ae7c9-Paper-round2.pdf

[42] Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4902–4912.

[43] Suchi Saria and Adarsh Subbaswamy. 2019. Tutorial: safe and reliable machine learning. *arXiv preprint arXiv:1904.07204* (2019).

[44] Mark Sendak, Madeleine Clare Elish, Michael Gao, Joseph Futoma, William Ratliff, Marshall Nichols, Armando Bedoya, Suresh Balu, and Cara O'Brien. 2020. " The human body is a black box" supporting clinical decision-making with deep learning. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 99–109.

[45] Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D. Sculley. 2017. No Classification without Representation: Assessing Geodiversity Issues in Open Data Sets for the Developing World. In *NIPS 2017 workshop: Machine Learning for the Developing World*.

[46] Liwei Song and Prateek Mittal. 2021. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2615–2632.

[47] Hendrik Strobelt, Daniela Oelke, Bum Chul Kwon, Tobias Schreck, and Hanspeter Pfister. 2015. Guidelines for effective usage of text highlighting techniques. *IEEE transactions on visualization and computer graphics* 22, 1 (2015), 489–498.

[48] Harini Suresh, Steven R. Gomez, Kevin K. Nam, and Arvind Satyanarayan. 2021. Beyond Expertise and Roles: A Framework to Characterize the Stakeholders of Interpretable Machine Learning and Their Needs. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 74, 16 pages. https://doi.org/10.1145/3411764.3445088

[49] Harini Suresh, Kathleen M Lewis, John Guttag, and Arvind Satyanarayan. 2022. Intuitively assessing ml model reliability through example-based explanations and editing model inputs. In *27th International Conference on Intelligent User Interfaces*. 767–781.

[50] Harini Suresh, Rajiv Movva, Amelia Lee Dogan, Rahul Bhargava, Isadora Cruxên, Ángeles Martinez Cuba, Guilia Taurino, Wonyoung So, and Catherine D'Ignazio. 2022. Towards Intersectional Feminist and Participatory ML: A Case Study in Supporting Feminicide Counterdata Collection. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 667–678.

[51] Nithum Thain, Lucas Dixon, and Ellery Wulczyn. 2017. Wikipedia Talk Labels: Toxicity. (2 2017). https://doi.org/10.6084/m9.figshare.4563973.v2

[52] Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. 2021. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *arXiv preprint arXiv:2106.00545* (2021).

[53] Martin Wattenberg and Fernanda B Viégas. 2008. The word tree, an interactive visual concordance. *IEEE transactions on visualization and computer graphics* 14, 6 (2008), 1221–1228.

[54] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).

[55] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2019. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 747–763.

[56] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. 2020. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 547–558.