Clarify: Improving Model Robustness With Natural Language Corrections

Yoonho Lee
yoonho@cs.stanford.edu
Stanford University
USA

Michelle S. Lam
mlam4@cs.stanford.edu
Stanford University
USA

Helena Vasconcelos
helenav@cs.stanford.edu
Stanford University
USA

Michael S. Bernstein
msb@cs.stanford.edu
Stanford University
USA

Chelsea Finn
cbfinn@stanford.edu
Stanford University
USA

ABSTRACT

The standard way to teach models is by feeding them lots of data. However, this approach often teaches models incorrect ideas because they pick up on misleading signals in the data. To prevent such misconceptions, we must necessarily provide additional information beyond the training data. Prior methods incorporate additional instance-level supervision, such as labels for misleading features or additional labels for debiased data. However, such strategies require a large amount of labeler effort. We hypothesize that people are good at providing textual feedback at the concept level, a capability that existing teaching frameworks do not leverage. We propose Clarify, a novel interface and method for interactively correcting model misconceptions. Through Clarify, users need only provide a short text description of a model's consistent failure patterns. Then, in an entirely automated way, we use such descriptions to improve the training process. Clarify is the first end-to-end system for user model correction. Our user studies show that non-expert users can successfully describe model misconceptions via Clarify, leading to increased worst-case performance in two datasets. We additionally conduct a case study on a large-scale image dataset, ImageNet, using Clarify to find and rectify 31 novel hard subpopulations.

CCS CONCEPTS

• **Human-centered computing** → **Natural language interfaces**; *Graphical user interfaces*; *Text input*; • **Computing methodologies** → **Learning from critiques**.

KEYWORDS

Interactive Model Correction, Natural Language Feedback, Human-in-the-Loop Machine Learning, Labeling Efficiency, Dataset Bias, Fairness, Model Robustness

UIST '24, October 13–16, 2024, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0628-8/24/10
https://doi.org/10.1145/3654777.3676362

ACM Reference Format:
Yoonho Lee, Michelle S. Lam, Helena Vasconcelos, Michael S. Bernstein, and Chelsea Finn. 2024. Clarify: Improving Model Robustness With Natural Language Corrections. In *The 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24), October 13–16, 2024, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3654777.3676362

1 INTRODUCTION

Machine learning systems trained with supervised learning often learn high-level misconceptions. For example, an image classifier trained to recognize birds may erroneously rely on background features like water rather than the visual appearance of the bird itself. Such misconceptions can cause unexpected failures when the model is deployed in new environments, leading to poor performance on specific subpopulations [3, 27, 32, 45, 49, 77]. These misconceptions arise because models are trained to extract correlations from the training data, which may contain spurious or misleading signals. Identifying such failure modes in advance is challenging due to the vast space of possible misconceptions. Left unaddressed, such misconceptions can repeatedly cause the model to make similar errors, significantly degrading real-world performance.

Existing methods have sought to mitigate misconceptions by providing additional supervision beyond the training data. Prior methods incorporate additional annotations about the spurious features, such as separate group indices indicating whether a bird image contains water, to encourage the model to ignore the spurious feature [72]. Alternatively, one can collect additional labeled data from a debiased distribution, for example, carefully curating images so that bird species is not correlated with the background [39]. A common theme in these approaches is that they require extensive human involvement in the form of additional instance-level supervision: in these approaches, additional annotations are needed at a scale comparable to that of the original training data. This makes these strategies prohibitively costly for settings where the original training data is already close to the full annotation budget. This is especially true in scenarios such as interactive machine learning [22, 24], rapid model correction, or data-driven exploration.

We posit that far less supervision suffices if we provide targeted feedback at the level of *concepts* rather than instances. Targeted feedback is a cornerstone for robustness in various contexts outside teaching machine learning models. Psychological studies underscore the pivotal role of corrective feedback in enhancing learning and decision-making [4, 31, 33, 40]. In causal inference, targeted

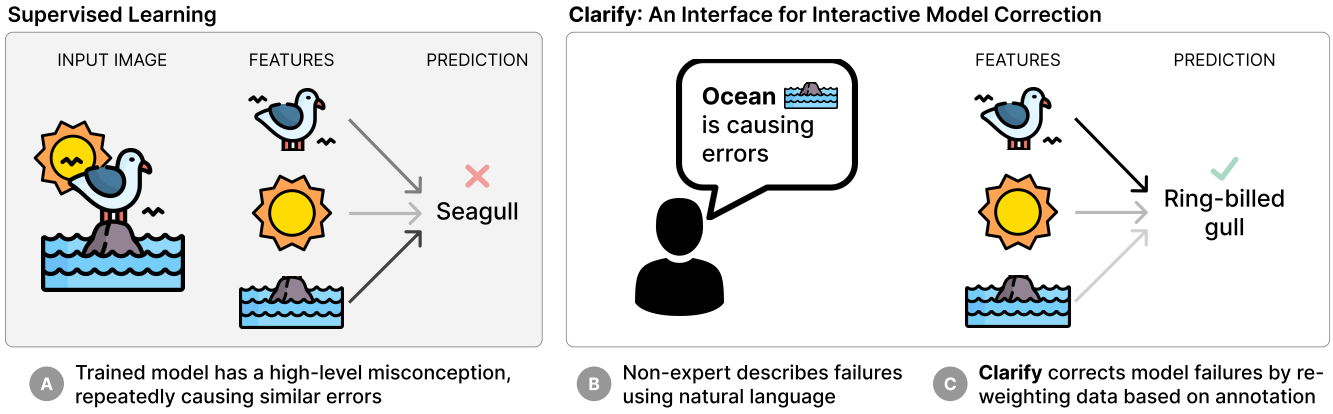

Figure 1: Clarify is an interface for interactively correcting model failures due to spurious correlations. (a) Given a model trained with supervised learning, (b) a human describes consistent failure modes of the model entirely in natural language. (c) We automatically incorporate these descriptions to improve the training process by reweighting the training data based on image-text similarity.

interventions allow us to identify causal effects, going beyond the limitations of observational studies, which can only capture correlations [64, 71, 74]. Despite such insights, existing forms of annotation for robustness in supervised learning fall short in this regard: they lack the specificity of targeted feedback and are provided without knowledge of the actual behavior of naively trained models. Through a lifetime of speaking and writing, people are highly adept at thinking and communicating at higher levels of abstraction. However, existing frameworks for teaching models are not adequately designed to leverage people's ability to provide concept-level feedback. This paper proposes a specific form of targeted feedback that aligns with these principles: natural language descriptions of model misconceptions.

We introduce Corrective Language Annotations for Robust InFerence (Clarify), a novel system that allows users to interactively correct failures of image classifiers using natural language. We consider image classifiers obtained by fine-tuning pre-trained models such as CLIP [67]. Although such classifiers achieve high average performance on held-out data, they often still suffer from high-level misconceptions. Clarify consists of an interface for collecting human feedback and a method for automatically incorporating this feedback to improve the training process. During interactions with the system, users observe a trained model's predictions on a held-out dataset and write short text descriptions that identify consistent failure modes. For instance, for a bird classifier relying on a spurious correlation between bird species and their backgrounds, a user might succinctly write that the model is mistakenly focusing on the "water background". We note that our system diverges substantially from standard supervised learning: we collect annotations *after* initial training and use these annotations in an entirely automated way to re-train the model based on the feedback. Please refer to Figure 1 for an overview of Clarify in relation to traditional supervised learning, and Figure 2 for a visualization of key interface features.

We instantiate Clarify in a web app implementation to carry out non-expert user studies (N=26) and evaluate the gathered feedback in addition to re-trained models. We find that within just a few minutes of interaction, non-expert users could use Clarify to identify consistent failure modes of models trained with standard supervised learning. Incorporating this feedback into the training process yields a statistically significant improvement in robustness: an average 17.1% increase in the accuracy of the worst-performing subpopulations. To further explore the ceiling of performance gains with Clarify, we perform a case study on a large and diverse dataset, ImageNet, using an expert annotator. This case study goes beyond standard datasets for spurious correlations with known failure modes and entails discovering and correcting previously unknown issues in a public dataset. We were able to identify 31 novel hard subpopulations in the dataset. We leveraged this information to improve the average worst-case accuracy across these subpopulations from 21.1% to 28.7% with only a 0.2% drop in average accuracy. With Clarify, we demonstrate that non-expert users can train and correct models by directly talking with them—opening up new design space for more efficient and accessible ways to design machine learning systems.

2 RELATED WORK

Our work draws upon literature in machine learning and human-computer interaction on strategies to efficiently correct machine learning models—whether to reduce training and annotation effort, bolster model robustness, or combat harmful failures.

2.1 ML Perspectives on Model Correction

Model correction methods in the machine learning literature tend to focus on developing novel algorithms while leaving user-facing processes intact, primarily focusing on using available labeled data more effectively.

Teaching ML models. As machine learning models require more and more resources to train, it becomes increasingly important to optimize the training process. The *machine teaching* literature aims to formalize the optimal training set for a given task and characterize its training complexity. While well-studied [18, 28, 54, 80, 97, 98], its application to large-scale models has been limited,

likely due to the substantial annotation burden required to teach a model from scratch.

Supervised learning, the dominant paradigm for training task-specific models, requires explicit labels for each instance and shows diminishing returns from additional human effort. Although active learning methods aim to reduce this annotation burden by selecting the most informative datapoints for labeling [47, 76], they still require humans to label individual datapoints. Our work proposes a new form of supervision that can rectify spurious correlations in labeled datasets: natural language descriptions of model errors. This form of supervision operates at a higher level of abstraction, providing a more efficient way to teach models with minimal additional annotation effort.

Robustness to spurious correlations. Models trained with standard supervised learning often exhibit a bias towards shortcut features—simple features that perform well on the training distribution yet fail to capture the underlying causal structure [3, 27, 65, 77]. Recent works have proposed methods to mitigate this issue, such as learning multiple functions consistent with the data [46, 62, 84, 85] and reweighting instances to render shortcut features non-predictive [13, 39, 57, 72]. However, these approaches often entail significant overhead for additional supervision, such as group labels indicating spurious features or carefully curated data free of spurious correlations. In contrast, Clarify requires only a few natural language descriptions of model errors, which are substantially easier to collect, rendering it especially practical for addressing misconceptions in large datasets.

Discovering failure modes. Our work builds upon a growing body of literature aimed at identifying and correcting model failure modes. Previous works discover poorly-performing subsets of data [5, 11, 16], devise methods to rectify specific failures [34, 55, 73, 94], or perform counterfactual data augmentation to penalize model reliance on erroneous features [37, 70, 87, 88, 92]. More closely related to our work are methods that leverage vision-language models to describe failure modes with natural language [20, 21, 38, 58, 90, 96]. Natural language descriptions of error slices have the advantage of being interpretable and naturally grounded in human understanding. However, many of the descriptions generated by these fully automated methods do not correspond to true model failures. For example, Zhang et al. [96] reports that DOMINO [21] can make nonsensical descriptions such as "mammoth" for a bird classification task. Our approach avoids such errors by incorporating humans in the loop, making it possible to discover spurious correlations in large datasets such as ImageNet.

2.2 Interactive Approaches to Model Correction

Meanwhile, the HCI literature tends to approach model correction by leaving existing algorithms largely intact, but amplifying user involvement through new interactions and visualizations.

Making ML models more accessible. The concept of a "low threshold" motivates HCI research on building systems accessible to non-experts [56]. Many works have specifically focused on lowering the bar for end-users' participation in various stages of creating and using machine learning models. Prior works have built tools for end-user data exploration [43, 63, 86], labeling [26, 69], feature selection [19, 22], model training [9, 24, 29, 44, 52, 53, 59], prompt engineering [36, 83, 91], and model auditing [7, 8, 15, 17, 41, 78, 82, 89, 93]. In line with this rich literature, our work aims to enable non-expert end users to correct high-level misconceptions in machine learning models. Since there is often a high effort barrier for users to engage in model development, we sought to demonstrate the efficacy of our approach even for limited amounts of user input. To our best knowledge, Clarify is the first to enable non-experts to use natural language to improve models in a fully end-to-end manner.

Interactive ML. The field of interactive machine learning (IML) demonstrated that by engaging users in the model development process through interactive labeling, users could rapidly develop models that better aligned with their needs [2, 19, 22, 25]. Subsequent work on interactive machine teaching (IMT) further explored how users could act as teachers rather than just low-level data labelers [68, 79]. Both of these literatures have explored how to instantiate high-level *concepts* with user-selected examples and demonstrations as well as predefined features and knowledge bases [6, 23, 24, 68]. Building on this body of work, we find that concepts are a helpful level of abstraction for non-technical users, as they align well with how users tend to decompose and communicate knowledge [60]. Clarify goes further by allowing users to specify arbitrary concepts in natural language to repair existing image classifiers that *do not* already use concept-based abstractions.

Eliciting high-level concepts. In our view, the most closely related works are those that elicit high-level concepts from humans [42, 81]. However, a key difference between these works and ours is that we focus on *negative knowledge*—teaching the model what not to learn—as opposed to these works, which specify what features the model should use. Especially for intuitive tasks like image classification, user knowledge is often *tacit* rather than explicit, making it hard to specify precisely [66]. Thus, it is easier for annotators to describe the failures of an existing model rather than define the desired behavior upfront. Restricting the feedback to negative knowledge is also important for scalability, as it is much easier to identify a few failure modes in an otherwise well-performing model than to specify the full set of useful concepts. This scalability is crucial for correcting spurious correlations in large-scale datasets such as ImageNet.

3 CLARIFY: A NATURAL LANGUAGE INTERFACE FOR MODEL CORRECTION

We now describe Corrective Language Annotations for Robust InFerence (Clarify), a novel system for identifying and mitigating spurious correlations in models trained with supervised learning. The main idea behind Clarify is to allow users to provide targeted natural language feedback to a model, helping the model focus on relevant features and ignore spurious ones. We employ a natural language interface to facilitate this process, which we describe in detail in this section. First, we describe the problem setting in Section 3.1. We then describe a concrete example of an interaction with the interface in Section 3.3, and two methods for incorporating this feedback into the training process in Section 3.4.

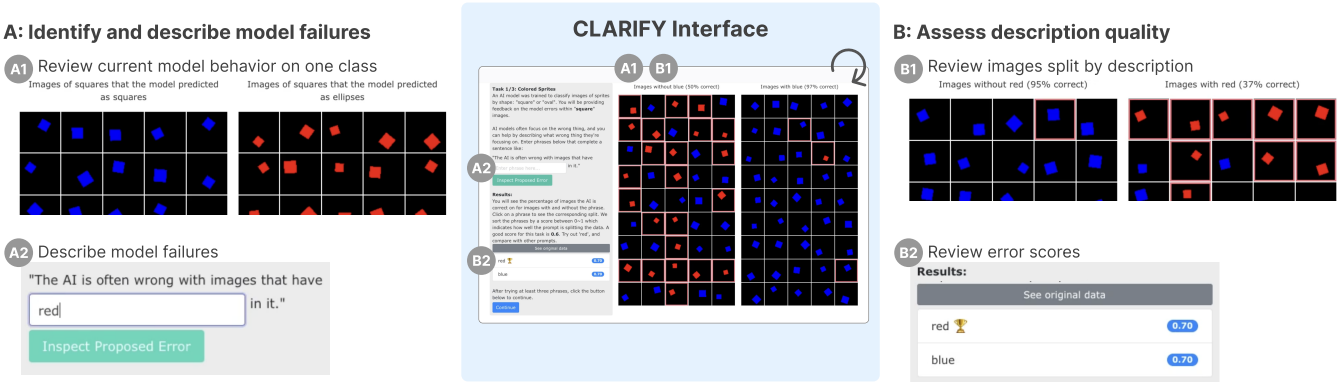

Figure 2: The Clarify interface enables users to iteratively (A) identify and describe model failures and (B) assess the quality of these descriptions. Users can review image examples of correct and incorrect predictions on one class, such as "square" (A1). Based on observed differences, they can input short, natural language descriptions of model failures, such as "red" squares (A2). The system surfaces feedback by splitting the data using the provided description (B1) and displaying an error score (B2). Users can repeat the process to generate improved descriptions.

3.1 Supervised Learning Problem Setup

We consider a standard supervised learning setting, where we are given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ of N labeled samples. Each label y_i belongs to one of C different classes: $y_i \in \{1, \ldots, C\}$. A model is trained to minimize the average loss across the training set, i.e., $\frac{1}{N}\sum_{i=1}^N \ell(f(x_i;\theta), y_i)$, where ℓ is a pointwise loss function such as cross-entropy, f is the model, and θ denotes model parameters. However, the dataset may inadvertently contain spurious correlations that hinder the model's ability to generalize to new distributions. To formalize spurious correlations, we can consider an extended dataset that includes an unknown attribute s_i for each instance, resulting in $\{(x_i, y_i, s_i)\}_{i=1}^N$ where $s_i \in \{1, \ldots, S\}$. For example, for a task where the labels y_i are bird species, the spurious attributes s_i could correspond to the background of the image x_i, which would be easier to infer from the input than the true label (i.e., bird species). A model trained on $\mathcal{D}$ may learn to rely on s_i to make predictions, thereby failing on new distributions where the previous correlation between s_i and y_i no longer holds. In general, we do not have annotations for these spurious attributes s_i or even know what they are in advance. Our goal is to correct the model's reliance on these spurious attributes without knowing a priori what they are.

3.2 Measuring Image-Text Similarity

To describe spurious attributes given only class-labeled image data, we leverage the capabilities of multimodal models such as CLIP [67], which encodes images and text into a shared embedding space. For a given image input I and text input T, CLIP outputs representations from separate vision and language branches, $e_i = f_i(I)$ and $e_t = f_t(T)$, respectively. This model is trained to maximize the similarity between the image and text representations for corresponding image-text pairs and minimize it for non-corresponding pairs through a contrastive loss function. We can estimate the similarity between a pair of image and text inputs by computing the cosine similarity of their respective representations:

$$\text{sim}(I, T) = \frac{e_i \cdot e_t}{\|e_i\|\|e_t\|}. \quad (1)$$

This black-box similarity function allows us to determine the relevance of a given image and text pair. The next section describes how Clarify leverages this relevance function to mitigate spurious correlations based solely on natural language feedback on a labeled validation set.

3.3 Interaction Workflow

To demonstrate how Clarify enables non-expert users to correct model misconceptions, we will walk through a user's workflow with the system (Figure 2). We will use a running example of a model trained to classify images of sprites as squares or ovals but mistakenly focuses on color rather than shape.

Reviewing model behavior. First, the user is presented with a summary view of the model's current behavior. The goal of this interface is to scaffold the user in rapidly identifying reasons underlying model failures. Drawing from a validation dataset, we display one class at a time (i.e., images of squares) and divide the examples into those that the model correctly classified (i.e., images classified as squares) on the left versus those that it incorrectly classified (i.e., images classified as ovals) on the right (Figure 2, A1). By presenting the images in this way, Clarify streamlines the user's task to one of identifying differences between sets. In our example, all of the images on the page are indeed squares, but the model is only making accurate predictions for the examples on the left and not those on the right. Comparing the images on the two sides, the user notices that the correct cases contain blue squares while the incorrect cases contain red squares.

Describing model failures. Now that the user has an initial idea of the model's misconception, they are tasked with describing this failure mode. Our system accepts short natural language descriptions of model failures (Figure 2, A2). In particular, users are asked to complete the following fill-in-the-blank sentence: "The

Phrase Category	Waterbirds	CelebA
Best WGA (per-user)	a bird with no head or as landbirds and a red outline, a blurry vision and they don't look like real birds, artic birds, beak, bird swims water, dark backgrounds and tall trees, forest, forest, forest, forest, forest, forests, forests, grass, greenery, landscape, landscapes, leaves, no water, plants, red, sandy beaches, seagulls, seagulls, trees, water, water	any other hair color than blonde or light hair color, backgrounds, bleach blonds, brown hair, buns, curls, curly hair, dyed hair, females, glasses, light background, light colors, men, men, men or short hair, older women, orange hair, pink, red, red, short hair, short hair, short haired men, smiles, white backgrounds, women
Best Error Score (per-user)	artic birds, birds, dark backgrounds and tall trees, ducks, forest, forest, forest, forest, forest, forests, forests, greenery, landscape, motion blur or can't make out a real bird, plants, sandy beaches, seagulls, trees, trees, trees, trees, water, water, water, waterfowl, wings	any other hair color than blonde or light hair color, bleach blonds, blond highlights, brown, darker blond hair, darker blonde, darker blonde hair, darker than blond, females, grey, males, men, men, men, men, men, men, men, men, men, men, men, pink, short white hair, very short hair, white, white, white
All Others	a lot of dark colors and no blue water, a lot of tree trunks, aqua blue water, been generated by ai, bird, bird wading in water, birds, birds floating, birds floating in water, birds standing in water, birds water, black, blue, blue, branches, branches, dark backgrounds, dark backgrounds and small birds, dark colors, darker backgrounds and a lot of trees, extended wings, eyes, flightless birds, flowers, game birds, grass, green, green, green, green, green plants, humans, land, landscapes, length of leg, lots of tree trunks, more dark colors than light colors and a lot of trees, mountains, no water, no water, no water and dark backgrounds, ocean coasts, people, people, people, plants, reeds, seagulls, shadows, sticks, tree trunk, trees, trees, trees, trees, trees, trees, very dark backgrounds and a lot of trees, water plants, wings, woods	bad lighting, bangs, beards, black hair, blue, blue, blue background, blue or black, brown, brown or dark hair, dark hair, darker hair, dim lighting, fair hair, flaxen, gold, golden hair, hair, hair, hats, hats, hats or bows, hazy, letters, light hair, little visible hair, long hair, males, males, males, men, more dark colors than light colors, non-blond hair. dark hair color. not blond, nondarkened hair, not blond, orange hair, people not facing the camera, red hair, red hair, redheads, short, short hair, short or curly hair, short or pulled back hair, shoulders, signs, skin color that is similar to their hair color, smiles, smiling faces, sunglasses, tan skin, teenagers, teeth, very tan skin, women

Table 1: The full set of model failure description phrases provided by non-expert annotators in our user study. The "Best WGA" and "Best Error Score" phrases were selected by identifying the phrase that achieved the highest Worst-Group Accuracy or Error Score, respectively, for each participant.

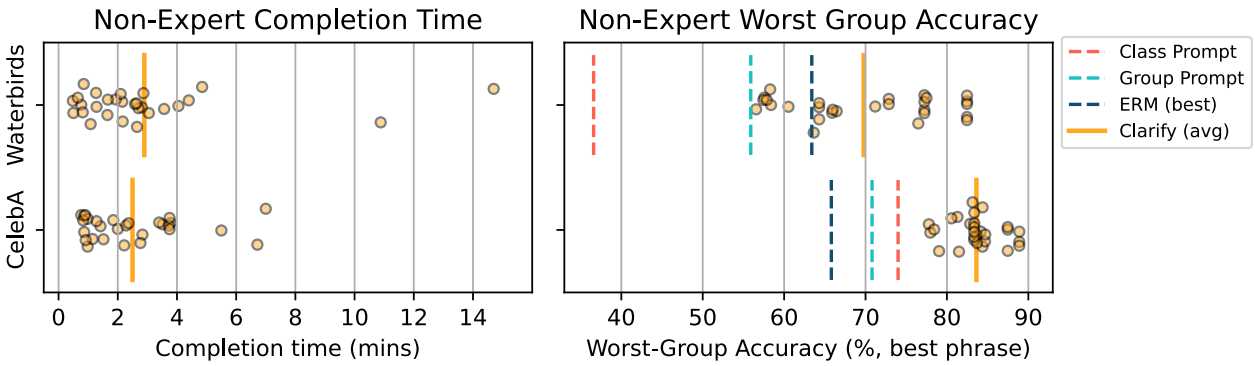

Figure 3: For both datasets, (left) non-experts completed annotation tasks using Clarify in less than 3 minutes on average, and (right) models retrained with non-expert annotations outperformed existing baselines in worst-group accuracy.

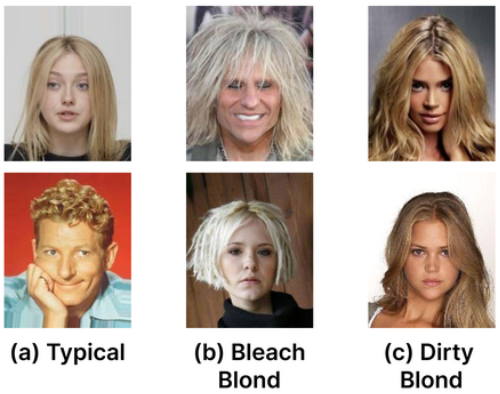

Figure 4: (a) Typical images from the "blond" class of CelebA. Non-experts provided textual feedback corresponding to hard subpopulations of (b) lighter and (c) darker hair colors.

AI is often wrong on images that have ___ in it." We find that this question is effective since users may not be familiar with the general concept of model failures or features. Continuing our running example, the user enters the phrase "red" here to describe what they observed.

Assessing descriptions. After the user submits their failure mode description, the Clarify interface helps them assess whether the description effectively describes the model's misconception. The system uses the CLIP model to compute the image-text similarity between each validation image and the user's description. Images with a similarity score above a threshold are considered to contain the feature described by the user. The interface presents a summary visualization that partitions the validation dataset based on this threshold, with matching images on the right and non-matching images on the left (Figure 2, B1). Additionally, we display a $0-1$ score that indicates how well the description separates the error cases from the correct predictions (Figure 2, B2). We note that

while the interface only shows *validation data* using the provided description, the user's natural language annotation will later be incorporated to partition the *training data* for model retraining.

Iterating on descriptions. As users may not be successful on their first attempt, Clarify aids users in iterating on their descriptions. Descriptions can fail for two reasons: (1) the description may be a valid differentiator, but may be modeled inaccurately due to the user's word choice and the limitations of CLIP-based similarity scoring, or (2) the description may not sufficiently differentiate the correct and incorrect cases. Clarify allows users to identify both of these failure modes. For example, the user can see if the model is not accurately identifying images with the color red based on the keyword "red" alone. In this case, they can experiment with alternate keywords to better isolate the difference, such as "red square" or "crimson". After iterating and isolating the red examples, the user can see if the provided score is still low, indicating that this description is not sufficient to repair model errors. With this information, users can revisit the original view and brainstorm additional descriptions, such as phrases related to the size or orientation of sprites.

We describe other details about the interface in Appendix A, including additional features that help users to refine their descriptions and assess their effectiveness. In Section 4.1, we evaluate the performance of non-expert annotators using Clarify and demonstrate that they can identify and describe model misconceptions.

3.4 Automatic Fine-Tuning

After collecting textual feedback from users, we incorporate this feedback into the training process for fine-tuning a foundation model. While the strategy below applies to any form of training, in this paper, we consider fine-tuning only the last layer on top of a pre-trained backbone network with frozen parameters. An error annotation is a tuple (c, T, τ), where c is the class label, T is the textual description, and τ is a threshold on the similarity function. Given such an error annotation, we partition the training data within class c into two subsets: $D_> = \{(x_i, y_i) \mid \text{sim}(x_i, T) > \tau\}$ and $D_< = \{(x_i, y_i) \mid \text{sim}(x_i, T) \leq \tau\}$. These two subsets correspond to images that are more and less similar to the provided text prompt, respectively, and serve as indicators of the model misconception identified by the annotator. Having identified these two subsets, we want to train a final model insensitive to the identified misconception, i.e., to achieve low training loss *without* using the feature that separates the two subsets.

We propose to use a simple distributionally robust optimization (DRO) objective function to achieve this goal. Having identified the two subsets $D_>$ and $D_<$, we propose to minimize the maximum loss over the two subsets to achieve robustness to the identified misconception; the loss function is given by:

$$\max\left(\mathcal{L}(f_\theta, D_>), \mathcal{L}(f_\theta, D_<)\right), \tag{2}$$

where $\mathcal{L}(f_\theta, D)$ is the average loss over the subset D. This objective ensures the model performs well on both subsets, avoiding the previous reliance on the spurious attribute. We optimize this objective using stochastic gradient descent with the max operator computed for each minibatch. We use this objective to train the last layer on top of a frozen pre-trained backbone model. In Section 4, we will measure the effectiveness of this fine-tuning approach based on language feedback. We note that this stage is fully automated, and there are no additional hyperparameters to tune beyond what was in the original training process.

4 EVALUATION

First, we note that our setup diverges substantially from assumptions in traditional supervised learning. Clarify involves collecting annotations *after* an initial round of training, and these annotations consist of targeted concept-level feedback rather than model-agnostic instance-level feedback. We consider this deviation from the conventional setup necessary to efficiently address the challenge of learning robust prediction rules from observational data. We seek to empirically answer the following questions about the Clarify system for interactively correcting model errors:

(1) Can non-expert users use Clarify to identify and describe spurious correlations in models trained with supervised learning?
(2) Can Clarify discover and rectify novel spurious correlations in large datasets such as ImageNet?
(3) How does Clarify compare to various automated methods which do not require human feedback?

For detailed experimental setup including datasets, models, and Prolific participants, see Appendix B.

4.1 User Study: Non-Expert Annotators Can Describe Model Errors

Identifying and annotating spurious correlations is a more nuanced task than conventional forms of annotation such as class labeling. This raises the question of whether non-expert annotators can perform this task, and if so, how efficiently they can do so. To answer these questions, we conduct a user study (N=26) to assess the ability of non-expert users to identify and describe spurious correlations in models trained with supervised learning (see Appendix B for study details).

In this study, we asked each participant to interact with models trained on the Waterbirds and CelebA datasets using the Clarify interface. In addition to qualitatively examining the feedback provided by participants, we evaluate the performance of the models trained on the feedback. We measure the robustness of these re-trained models using the worst-group accuracy (WGA) metric, which measures the accuracy of the worst-performing subpopulation in the dataset. For example, in the Waterbirds dataset, the worst-group accuracy is the minimum average accuracy across the four subpopulations ("landbird in land", "landbird in water", "waterbird in land", and "waterbird in water").

We summarize the results of our user study in Figure 3 and Table 3, comparing to zero-shot prompting and fine-tuning baselines. Users were able to achieve these performance improvements with minimal annotation effort, averaging 2.7 minutes (SD=2.5) per dataset. As a point of comparison, Chang et al. [10] found that annotators required 4.4 and 10.2 minutes to provide high-quality labels for 100 examples in the much simpler MNIST and K-MNIST datasets, respectively. For the best-performing annotation from each user, the average worst-group accuracy was 69.8 (SD=9.0, max=82.5) for Waterbirds and 83.6 (SD=3.1, max=88.9) for the CelebA dataset. Single-factor ANOVAs showed a statistically significant increase in worst-group accuracy from using Clarify: $F(1, 24) = 12.96, p < 0.002$ for

Waterbirds and $F(1, 24) = 262.44, p < 0.001$. We then conducted Tukey HSD posthoc tests to compare conditions between all pairs of methods. This test confirmed statistical significance at the $p < .005$ level for all pairwise between Clarify and the baselines for both datasets. In summary, non-expert annotations using Clarify significantly outperformed the baseline methods we considered.

We additionally find that non-expert annotators identified previously unknown spurious correlations in the CelebA dataset. In addition to the known spurious correlation between hair color and gender, participants identified subpopulations of "dirty blonde" and "bleach blond" individuals, which models consistently misclassified (Figure 4). Our findings suggest that Clarify can enable non-expert annotators to identify and describe model misconceptions. This opens up the possibility of leveraging a broader workforce for interactively improving models trained on web-scale datasets such as ImageNet or LAION [14, 75].

4.2 Case Study: Discovering and Mitigating Model Misconceptions in ImageNet

Class Name	Spurious Feature
cup	tea cup
weasel	snow weasel
wool	yarn ball
space bar	computer mouse
letter opener	silver
loupe	person holding a magnifying glass
mouse	desk and laptop
bakery	store front
sunscreen	person with sunburns
minivan	black minivan
plate rack	machine
briard	shaggy dog
lens cap	camera equipment
bighorn	rocky hillside
mushroom	red
rifle	wooden barrel
spotlight	shining
chocolate sauce	pastries with chocolate
terrapin	pond
sidewinder	sand
bikini	group of people
flatworm	coral reef
monitor	monitor on a desk
breastplate	museum display
projectile	rocket in a building
academic gown	many people in robes
velvet	pink velvet
bathtub	person
sliding door	car
partridge	tall grass
ear	green

Table 2: The 31 identified spurious features in the ImageNet dataset. All annotation was performed on the validation split.

We now evaluate whether Clarify can be used to discover novel model misconceptions in models trained on the ImageNet training set. It is important to develop tools to find consistent errors and methods to mitigate them for such widely used large-scale datasets. For this evaluation, the authors of this paper use the Clarify interface for ImageNet and additionally evaluate whether the resulting annotations can improve model robustness.

Identified subpopulations. Using Clarify, we identified 31 model misconceptions in ImageNet; we show a full list in Section 4.2 and visualize many of them in Figure 5 Despite ImageNet being a widely studied dataset, to our best knowledge, no prior works have identified these misconceptions. For example, we visualize a spurious correlation in the "sliding door" class in Figure 6: sliding doors are negatively correlated with cars in the training set, causing standard models to misclassify cars with sliding doors. We further evaluate the performance of a standard ImageNet model on each identified minority and majority split. Results in Figure 11 show that a naively fine-tuned ERM model consistently underperforms on the minority split for each class, indicating that standard models indeed rely on each of these spurious correlations. This trend continues to hold on ImageNet-V2, which follows a different distribution from the validation set we use to collect feedback.

Fine-tuning while avoiding spurious correlations. We use the collected annotations to fine-tune a model on ImageNet and evaluate this fine-tuned model on various splits of the ImageNet validation set. Results in Figure 8 show that the retrained model achieves higher minority split performance on many classes. Aggregate metrics in Figure 7 show that fine-tuning with Clarify annotations improves the average minority-split accuracy from 21.1% to 28.7%, with only a 0.2% drop in overall average accuracy. We emphasize that no additional data was used during fine-tuning—the annotations from Clarify were only used to find a better reweighting of the same training data used to train the original ERM model.

4.3 Technical Evaluation

We compare Clarify with various automated methods for handling model misconceptions. We note that none of the following comparisons are apples-to-apples: Clarify operates in a new problem setting that involves targeted human feedback, while the automated methods are designed to use pre-existing data, annotations, or models. Our goal is to see how the new form of targeted supervision in Clarify compares to prior approaches since it is substantially more information-dense and expressive and thus scalable to large datasets.

Comparison with zero-shot prompting methods. We compare the re-trained classifier using Clarify annotations with zero-shot prompting methods in Table 4. Clarify shows substantially better worst-group accuracy and robustness gap on the Waterbirds and CelebA datasets. Among these points of comparison, RoboShot [1] is notable as it is an automated method that leverages state-of-the-art foundation models [12, 35, 61]. We note that this is not necessarily a fair comparison in either direction: RoboShot uses a powerful language model to alter its prompts, whereas Clarify leverages targeted human feedback. Nevertheless, this comparison is still informative in that it shows that we can get much more

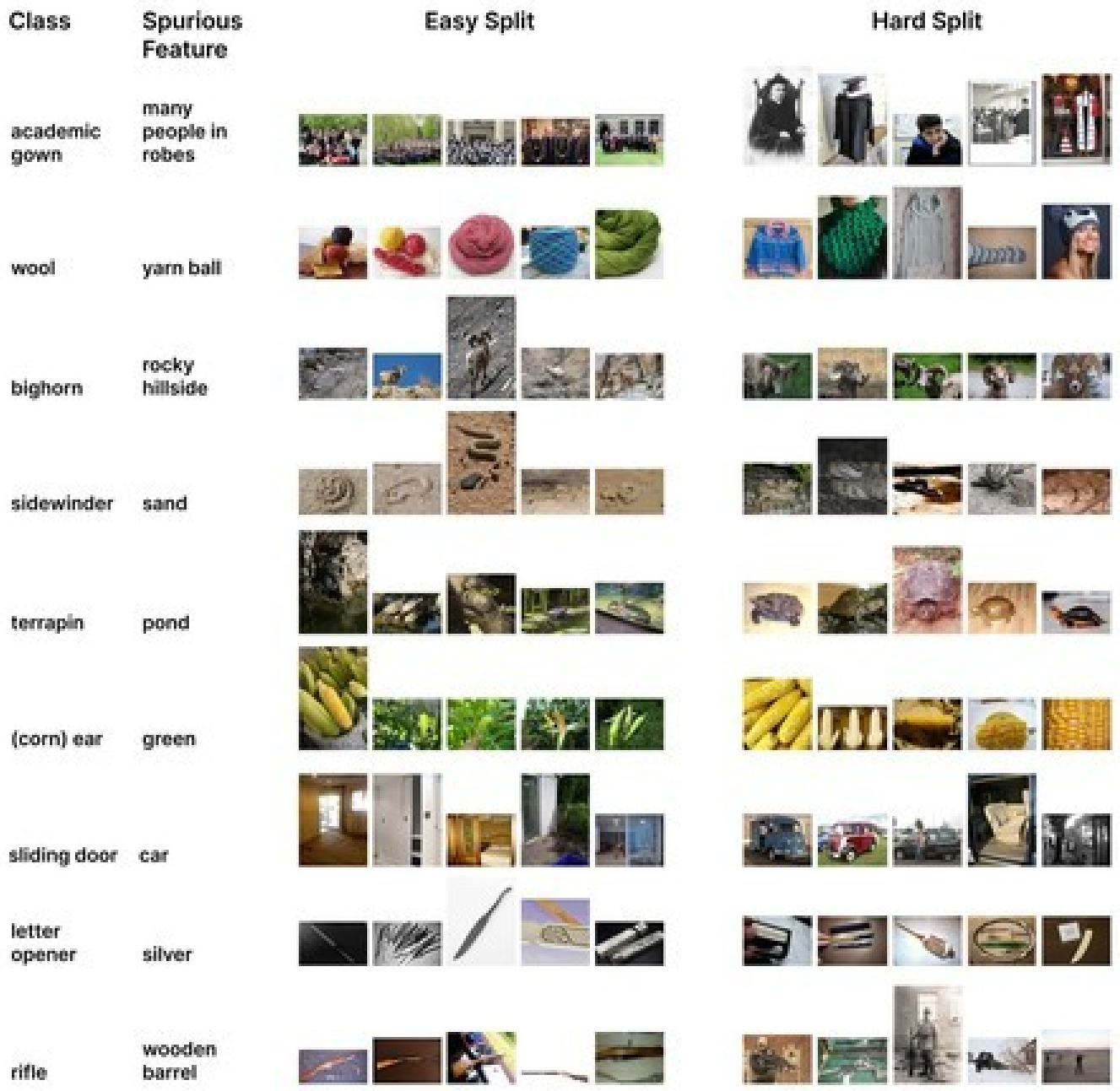

Figure 5: Representative samples corresponding to nine identified spurious correlations in ImageNet. All images shown are in the ImageNet validation set, and belong to the class shown in the first column. Similarity to the specified text annotation splits separates the "easy" and "hard" examples.

leverage out of natural language feedback by having it directly address gaps in existing training data.

Comparison with methods for spurious correlations. We assess how re-training a model with expert annotations from Clarify compares to existing automated methods for addressing spurious correlations. We compare with representative prior methods, which similarly fine-tune CLIP models and/or reweight training data. In addition to Clarify, we evaluate zero-shot CLIP [67] with class-based and group-based prompts, DFR [39], and Group DRO [72]. We describe experimental details for each method in Appendix B.

Figure 6: An example of a spurious correlation found on ImageNet. Within the "sliding door" class, the model successfully classifies (a) images of sliding doors inside buildings. However, it is wrong on all instances of (b) sliding doors on cars. This is one of the 31 spurious correlations we found; please refer to Figure 5 for more visualizations.

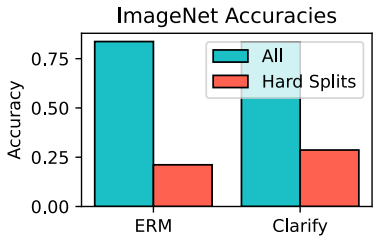

Figure 7: Average accuracies on ImageNet data. Fine-tuning with Clarify substantially improves accuracy on hard splits, while keeping overall accuracy intact.

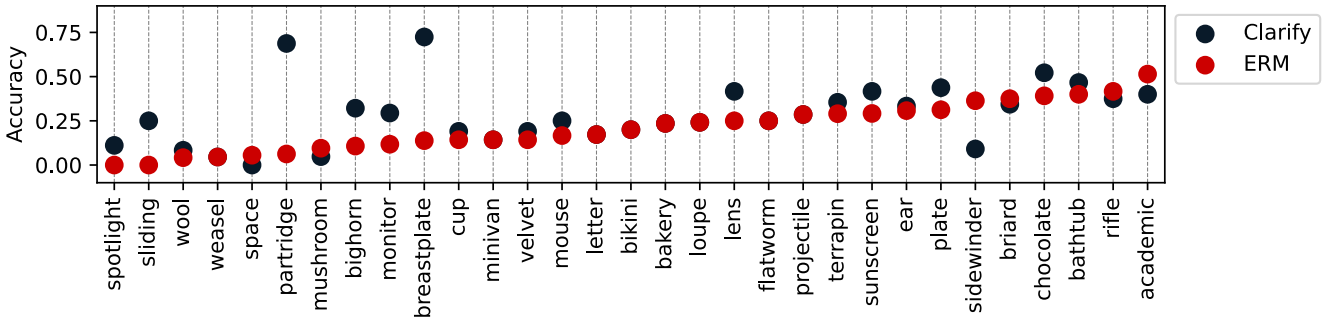

Figure 8: Average minority split accuracy for each of the 31 identified spurious correlations. Fine-tuning with textual feedback from Clarify improves minority split accuracy in many classes.

		Waterbirds			CelebA		
Data Assumptions	Method	WG	Avg	Gap	WG	Avg	Gap
Zero-Shot	Class Prompt	36.6	92.2	55.6	74.0	81.9	7.9
	Group Prompt	55.9	87.8	31.9	70.8	82.6	11.8
Labels	ERM	7.9	93.5	85.6	11.9	94.7	82.8
	ERM (ours)	63.4	96.0	32.6	31.1	95.4	64.3
	ERM (ours, class-balanced)	48.6	95.2	46.7	65.8	93.4	27.6
	ERM (ours, worst-class)	55.9	95.8	39.9	56.9	94.1	37.2
Labels, Text Feedback	Clarify (avg non-expert)	69.8	84.1	13.3	83.7	93.2	9.5
	Clarify (best non-expert)	**82.5**	90.7	8.2	88.8	92.9	4.1
	Clarify (author)	75.7	83.8	**8.1**	**89.1**	92.1	**3.0**
Labels, Group Labels	DFR (downsample)	63.9	91.8	27.9	76.9	92.5	15.6
	DFR (upsample)	51.3	92.4	41.1	89.6	91.8	2.2
	DFR (our implementation)	78.7	90.8	12.1	**90.6**	91.9	**1.3**
	Group DRO (our implementation)	**81.3**	88.1	**6.8**	89.2	91.8	2.7

Table 3: Evaluation of methods for group robustness using the CLIP-ResNet50 backbone. For Clarify, we show the average and best participant from our non-expert user study (N=26) in addition to feedback from an author of this paper. Fine-tuning with annotations from Clarify consistently outperforms methods that use only text (zero-shot) or label information. All results other than ours are from Zhang and Ré [95].

Our results on the Waterbirds and CelebA datasets, summarized in Table 3, demonstrate that Clarify consistently outperforms approaches that use zero-shot prompts or class labels in terms of worst-group accuracy and robustness gaps and is competitive with specialized methods that use instance-level annotations for spurious attributes. We show extended results with an alternative network architecture as the pre-trained backbone in Table 6.

Moreover, the key advantage of Clarify is its scalability to large datasets, a feature that no prior automated method has demonstrated. Such scalability is crucial when applying these ideas to

Model	Method	Waterbirds			CelebA		
		Avg	WG(↑)	Gap(↓)	AVG	WG(↑)	Gap(↓)
ALIGN	Class Prompt	72.0	50.3	21.7	81.8	77.2	4.6
	Group Prompt	72.5	5.8	66.7	78.3	67.4	10.9
	RoboShot [1]	50.9	41.0	**9.9**	86.3	83.4	2.9
AltCLIP	Class Prompt	90.1	35.8	54.3	82.3	79.7	2.6
	Group Prompt	82.4	29.4	53.0	82.3	79.0	3.3
	RoboShot [1]	78.5	54.8	23.7	86.0	77.2	8.8
CLIP (ViT-L/14)	Class Prompt	88.7	27.3	61.4	80.6	74.3	6.3
	Group Prompt	70.7	10.4	60.3	77.9	68.9	9.0
	RoboShot [1]	79.9	45.2	34.7	85.5	82.6	2.9
	Clarify	96.8	**81.8**	14.9	90.9	**88.8**	**2.1**

Table 4: Comparison with different zero-shot CLIP prompting strategies for group robustness. Fine-tuning with Clarify substantially outperforms RoboShot, a method that leverages state-of-the-art foundation models to automatically generate text prompts. All results besides ours are from Adila et al. [1].

real-world problems where the scale and diversity of data are ever-increasing.

Comparison with automated bias discovery. Since annotation time is a key cost of Clarify, we investigate whether such feedback brings concrete benefits over fully automated methods for discovering model biases. Specifically, we compare Clarify with Bias-to-Text [38], a representative method for automated bias discovery. We evaluated the automated Bias-to-Text pipeline on several classes in the ImageNet validation set, in which we identified spurious correlations. We find that Bias-to-Text can identify relevant keywords for each class, but it has specific pitfalls that make it difficult to use without human oversight. In Table 5, we show 10 keywords identified by Bias-to-Text for 6 classes in ImageNet. We note that the top identified keywords, i.e., the ones with the highest CLIP score, often describe something highly related to the class label, such as "goat" for the "bighorn". Additionally, we numerically compare the discovered annotations in their ability to improve model robustness in Figure 9. The annotations for Bias-to-Text show substantially higher minority split accuracy (Clarify 21.1%, Bias-to-Text 45.2%), with a smaller gap with the majority split. Furthermore, after re-training with these annotations, we observe a slight decrease in held-out minority split accuracy (45.2% to 44.3%). This is in contrast to re-training with Clarify annotations, which substantially improved minority split accuracy (21.1% to 28.7%). However, we note that automated discovery methods are highly useful in the context of Clarify, as they can prime annotators with a set of candidate keywords or help prioritize the most promising parts of the dataset.

5 DISCUSSION

Our evaluation demonstrates that Clarify enables non-expert users to identify and address misconceptions within machine learning models in an end-to-end manner. User feedback is immediately actionable and can be used to improve model robustness without any additional data collection. Allowing users to improve models end-to-end is important for both scalability and building public trust. The worst-case accuracy gains from re-training with Clarify annotations are substantial, and the method is competitive with or outperforms existing automated methods for addressing spurious correlations. Furthermore, we show that Clarify can be used to discover novel misconceptions from training on large-scale datasets such as ImageNet and that the resulting annotations can be used to improve model robustness.

5.1 Comparison to Supervised Learning

Our approach to teaching machines diverges substantially from standard supervised learning. We highlight two key properties of Clarify which offer complementary strengths to conventional supervised learning. First, we collect annotations *after* initial training, allowing the model's behavior to inform the annotation process. Interacting with a model after training enables our interface to specifically elicit *negative knowledge* from users, i.e., telling the model what *not* to focus on. People are often better at identifying errors than articulating complex rules, and negative knowledge can fill the gaps in the positive knowledge in the original labeled training set. Second, annotations from Clarify have a substantially higher information density than conventional forms of annotations. Unlike instance-specific labels, textual feedback encapsulates concept-level "global" insights applicable across the entire dataset, making it a more efficient mode of human-AI communication.

5.2 Limitations

However, we acknowledge certain limitations of Clarify in its current form. While we find that non-expert users can provide meaningful feedback on model errors, eliciting high-quality feedback from non-experts remains challenging. Their feedback can be too generic, e.g., "the model is wrong" or directly describing the class label rather than the spurious correlation; signaling the right level of granularity to the user is a challenge in this mode of interaction. We also find that some users could only find model errors that are very visually salient, such as the presence of a specific large object in the image. While many users provided useful feedback, some struggled, likely because they had no prior knowledge of the model's capabilities.

In general, Clarify can only help with model failures that can be concisely described in natural language. This excludes more complex failures, such as those requiring domain-specific knowledge or not easily articulated in text. Also, as much as Clarify benefits from using a pre-trained backbone model for bridging between natural language descriptions and images, it inherits the limitations of the pre-trained CLIP model. This includes biases in the training distribution and limited coverage of specialized domains such as medical imaging or scientific data.

5.3 Future Work

A natural extension of Clarify is to apply this framework of collecting textual feedback on model errors to large language models (LLMs). A key challenge in this direction is in designing effective user interfaces that allow users to quickly understand a model's overall behavior and develop strategies to elicit specific, actionable user feedback. Enabling this mode of interaction, where users can teach LLMs by critiquing their mistakes, could be a promising approach for making powerful models easier to align and adapt to specific user and community needs without requiring extensive annotation costs. Collecting feedback *after* initial model training is especially appealing in the context of current LLMs, where the full scope of a model's emergent behaviors can be difficult to predict in advance.

We are also excited about increasing expressivity by designing interfaces that allow users to provide more nuanced and open-ended feedback, potentially through extended text passages or interactive multi-turn dialogue with the model. Such interfaces could leverage more knowledge and expertise from people, including domain experts, which may be particularly valuable in specialized domains such as healthcare or scientific discovery. We intentionally scoped our evaluation to non-experts to demonstrate the broad applicability of our end-to-end approach. Our non-expert results establish a floor for improvement from the proposed workflow, and further iterations will only benefit from richer user feedback from users with more expertise. Allowing for richer feedback will also likely be critical in more complex tasks like open-ended instruction following and in data modalities such as video or audio, where the model's behavior is more complex and harder to summarize succinctly.

6 CONCLUSION

This paper introduces Clarify, a novel interface for correcting high-level model misconceptions in machine learning models. Clarify enables non-expert users to provide targeted feedback on model errors, which can be used to improve model robustness. We also show that Clarify can be used to discover novel misconceptions in large-scale datasets such as ImageNet. We believe that the general idea of correcting models with targeted textual feedback has the potential to substantially improve model performance while reducing the need for extensive manual annotation.

ACKNOWLEDGMENTS

We thank members of the IRIS lab and Pang Wei Koh for helpful discussions and feedback. YL acknowledges support from KFAS. This work was partly supported by NSF CAREER award 2237693 and Schmidt Sciences.

REFERENCES

[1] Dyah Adila, Changho Shin, Linrong Cai, and Frederic Sala. 2023. Zero-Shot Robustification of Zero-Shot Models With Foundation Models. *arXiv preprint arXiv:2309.04344* (2023).

[2] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine* 35, 4 (Dec. 2014), 105–120. https://doi.org/10.1609/aimag.v35i4.2513

[3] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International Conference on Machine Learning*.

[4] Robert L Bangert-Drowns, Chen-Lin C Kulik, James A Kulik, and MaryTeresa Morgan. 1991. The instructional effect of feedback in test-like events. *Review of educational research* 61, 2 (1991), 213–238.

[5] Yujia Bao and Regina Barzilay. 2022. Learning to split for automatic bias detection. *arXiv preprint arXiv:2204.13749* (2022).

[6] 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)*. 105–112. https://doi.org/10.1109/VAST.2015.7347637

[7] Ángel Alexander Cabrera, Abraham J. Druck, Jason I. Hong, and Adam Perer. 2021. Discovering and Validating AI Errors With Crowdsourced Failure Reports. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 425 (oct 2021), 22 pages. https://doi.org/10.1145/3479569

[8] Ángel Alexander Cabrera, Erica Fu, Donald Bertucci, Kenneth Holstein, Ameet Talwalkar, Jason I. Hong, and Adam Perer. 2023. Zeno: An Interactive Framework for Behavioral Evaluation of Machine Learning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>Hamburg</city>, <country>Germany</country>, </conf-loc>) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 419, 14 pages. https://doi.org/10.1145/3544548.3581268

[9] Michelle Carney, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. 2020. Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>Honolulu</city>, <state>HI</state>, <country>USA</country>, </conf-loc>) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382839

[10] Chia-Ming Chang, Xi Yang, and Takeo Igarashi. 2022. An Empirical Study on the Effect of Quick and Careful Labeling Styles in Image Annotation. In *Graphics Interface 2022*. https://openreview.net/forum?id=SDyj8aZBPrs

[11] Mayee Chen, Karan Goel, Nimit S Sohoni, Fait Poms, Kayvon Fatahalian, and Christopher Ré. 2021. Mandoline: Model evaluation under distribution shift. In *International conference on machine learning*. PMLR, 1617–1629.

[12] Zhongzhi Chen, Guang Liu, Bo-Wen Zhang, Fulong Ye, Qinghong Yang, and Ledell Wu. 2022. Altclip: Altering the language encoder in clip for extended language capabilities. *arXiv preprint arXiv:2211.06679* (2022).

[13] Elliot Creager, Joern-Henrik Jacobsen, and Richard Zemel. 2021. Environment Inference for Invariant Learning. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 2189–2200.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[15] Wesley Hanwen Deng, Boyuan Guo, Alicia Devrio, Hong Shen, Motahhare Eslami, and Kenneth Holstein. 2023. Understanding Practices, Challenges, and Opportunities for User-Engaged Algorithm Auditing in Industry Practice. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>Hamburg</city>, <country>Germany</country>, </conf-loc>) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 377, 18 pages. https://doi.org/10.1145/3544548.3581026

[16] Greg d'Eon, Jason d'Eon, James R Wright, and Kevin Leyton-Brown. 2022. The spotlight: A general method for discovering systematic errors in deep learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 1962–1981.

[17] Alicia DeVos, Aditi Dhabalia, Hong Shen, Kenneth Holstein, and Motahhare Eslami. 2022. Toward User-Driven Algorithm Auditing: Investigating users' strategies for uncovering harmful algorithmic behavior. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>New Orleans</city>, <state>LA</state>, <country>USA</country>, </conf-loc>) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 626, 19 pages. https://doi.org/10.1145/3491102.3517441

[18] Gregory Druck, Gideon Mann, and Andrew McCallum. 2008. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 595–602.

[19] 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.

[20] Lisa Dunlap, Clara Mohri, Devin Guillory, Han Zhang, Trevor Darrell, Joseph E Gonzalez, Aditi Raghunathan, and Anna Rohrbach. 2022. Using language to extend to unseen domains. In *The Eleventh International Conference on Learning Representations*.

[21] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. 2022. Domino: Discovering systematic errors with cross-modal embeddings. *arXiv preprint arXiv:2203.14960* (2022).

[22] 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.

[23] Rebecca Fiebrink, Dan Trueman, and Perry R. Cook. 2009. A Meta-Instrument for Interactive, On-the-fly Machine Learning. In *International Conference on New Interfaces for Musical Expression* (Pittsburgh, PA) *(NIME '09)*.

[24] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. 2008. CueFlik: interactive concept learning in image search. In *Proceedings of the sigchi conference on human factors in computing systems*. 29–38.

[25] Jules Françoise, Baptiste Caramiaux, and Téo Sanchez. 2021. Marcelle: Composing Interactive Machine Learning Workflows and Interfaces. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 39–53. https://doi.org/10.1145/3472749.3474734

[26] Irena Gao, Gabriel Ilharco, Scott Lundberg, and Marco Tulio Ribeiro. 2022. Adaptive testing of computer vision models. *arXiv preprint arXiv:2212.02774* (2022).

[27] 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.

[28] Sally A. Goldman and Michael J. Kearns. 1991. On the Complexity of Teaching. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory* (Santa Cruz, California, USA) *(COLT '91)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 303–314.

[29] Patrice Simard Jina Suh Gonzalo Ramos, Christopher Meek 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. https://doi.org/10.1080/07370024.2020.1734931 arXiv:https://doi.org/10.1080/07370024.2020.1734931

[30] Maarten Grootendorst. 2020. KeyBERT: Minimal keyword extraction with BERT. https://doi.org/10.5281/zenodo.4461265

[31] John Hattie and Helen Timperley. 2007. The power of feedback. *Review of educational research* 77, 1 (2007), 81–112.

[32] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261* (2019).

[33] Daniel R Ilgen, Cynthia D Fisher, and M Susan Taylor. 1979. Consequences of individual feedback on behavior in organizations. *Journal of applied psychology* 64, 4 (1979), 349.

[34] Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. 2022. Distilling model failures as directions in latent space. *arXiv preprint arXiv:2206.14754* (2022).

[35] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*. PMLR, 4904–4916.

[36] Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. 2022. Prompt-based prototyping with large language models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*.

[37] Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434* (2019).

[38] Younghyun Kim, Sangwoo Mo, Minkyu Kim, Kyungmin Lee, Jaeho Lee, and Jinwoo Shin. 2023. Explaining visual biases as words by generating captions. *arXiv preprint arXiv:2301.11104* (2023).

[39] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. 2022. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937* (2022).

[40] Avraham N Kluger and Angelo DeNisi. 1996. The effects of feedback interventions on performance: a historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological bulletin* 119, 2 (1996), 254.

[41] Michelle S. Lam, Mitchell L. Gordon, Danaë Metaxa, Jeffrey T. Hancock, James A. Landay, and Michael S. Bernstein. 2022. End-User Audits: A System Empowering Communities to Lead Large-Scale Investigations of Harmful Algorithmic Behavior. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2, Article 512 (nov 2022), 34 pages. https://doi.org/10.1145/3555625

[42] Michelle S Lam, Zixian Ma, Anne Li, Izequiel Freitas, Dakuo Wang, James A Landay, and Michael S Bernstein. 2023. Model Sketching: Centering Concepts in Early-Stage Machine Learning Model Design. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–24.

[43] Michelle S. Lam, Janice Teoh, James A. Landay, Jeffrey Heer, and Michael S. Bernstein. 2024. Concept Induction: Analyzing Unstructured Text with High-Level Concepts Using LLooM. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (<conf-loc>, <city>Honolulu</city>, <state>HI</state>, <country>USA</country>, </conf-loc>) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 766, 28 pages. https://doi.org/10.1145/3613904.3642830

[44] D. Lane. 2021. *Machine Learning for Kids: A Project-Based Introduction to Artificial Intelligence*. No Starch Press. https://books.google.com/books?id=g3ISEAAAQBAJ

[45] Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. 2021. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems* 34 (2021), 29348–29363.

[46] Yoonho Lee, Huaxiu Yao, and Chelsea Finn. 2022. Diversify and Disambiguate: Learning From Underspecified Data. *arXiv preprint arXiv:2202.03418* (2022).

[47] 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.

[48] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*. PMLR, 12888–12900.

[49] Zhiheng Li, Ivan Evtimov, Albert Gordo, Caner Hazirbas, Tal Hassner, Cristian Canton Ferrer, Chenliang Xu, and Mark Ibrahim. 2022. A Whac-A-Mole Dilemma: Shortcuts Come in Multiples Where Mitigating One Amplifies Others. (2022).

[50] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*. PMLR, 6781–6792.

[51] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[52] M. Matas, A. Menges, and M. Beissinger. 2020. Lobe: Deep Learning Made Simple. https://lobe.ai/ Accessed: 2024-06-08.

[53] Microsoft. 2023. Microsoft Azure Machine Learning Studio (classic). https://studio.azureml.net/ Accessed: 2024-06-08.

[54] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. 1003–1011.

[55] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309* (2021).

[56] Brad Myers, Scott E. Hudson, and Randy Pausch. 2000. Past, present, and future of user interface software tools. *ACM Trans. Comput.-Hum. Interact.* 7, 1 (mar 2000), 3–28. https://doi.org/10.1145/344949.344959

[57] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. 2020. Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems* 33 (2020), 20673–20684.

[58] Yannic Neuhaus, Maximilian Augustin, Valentyn Boreiko, and Matthias Hein. 2023. Spurious Features Everywhere - Large-Scale Detection of Harmful Spurious Features in ImageNet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 20235–20246.

[59] Felicia Ng, Jina Suh, and Gonzalo Ramos. 2020. Understanding and Supporting Knowledge Decomposition for Machine Teaching. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (<conf-loc>, <city>Eindhoven</city>, <country>Netherlands</country>, </conf-loc>) *(DIS '20)*. Association for Computing Machinery, New York, NY, USA, 1183–1194. https://doi.org/10.1145/3357236.3395454

[60] Felicia Ng, Jina Suh, and Gonzalo Ramos. 2020. Understanding and Supporting Knowledge Decomposition for Machine Teaching. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (Eindhoven, Netherlands) *(DIS '20)*. Association for Computing Machinery, New York, NY, USA, 1183–1194. https://doi.org/10.1145/3357236.3395454

[61] OpenAI. 2023. GPT-4 Technical Report. *ArXiv* abs/2303.08774 (2023).

[62] Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. 2022. Agree to Disagree: Diversity through Disagreement for Better Transferability. *arXiv preprint arXiv:2202.04414* (2022).

[63] PAIR People + AI Research Initiative. 2017. Facets: Know Your Data. https://pair-code.github.io/facets/ Accessed: 2024-06-08.

[64] Judea Pearl. 2009. *Causality*. Cambridge university press.

[65] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. 2021. Gradient Starvation: A Learning Proclivity in Neural Networks. In *Advances in Neural Information Processing*

Systems, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.).
[66] Michael Polanyi. 2009. The tacit dimension. In *Knowledge in organisations*. Routledge, 135–146.
[67] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
[68] 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.
[69] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB endowment. International conference on very large data bases*, Vol. 11. NIH Public Access, 269.
[70] Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E Peters, and Matt Gardner. 2021. Tailor: Generating and perturbing text with semantic controls. *arXiv preprint arXiv:2107.07150* (2021).
[71] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* 66, 5 (1974), 688.
[72] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731* (2019).
[73] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. 2021. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems* 34 (2021), 23359–23373.
[74] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proc. IEEE* 109, 5 (2021), 612–634.
[75] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems* 35 (2022), 25278–25294.
[76] Burr Settles. 2009. Active learning literature survey. (2009).
[77] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. 2020. The pitfalls of simplicity bias in neural networks. *Conference on Neural Information Processing Systems* (2020).
[78] Hong Shen, Alicia DeVos, Motahhare Eslami, and Kenneth Holstein. 2021. Everyday Algorithm Auditing: Understanding the Power of Everyday Users in Surfacing Harmful Algorithmic Behaviors. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 433 (oct 2021), 29 pages. https://doi.org/10.1145/3479577
[79] Patrice Simard, Saleema Amershi, Max Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Chris Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. 2017. *Machine Teaching: A New Paradigm for Building Machine Learning Systems.* Technical Report MSR-TR-2017-26. https://www.microsoft.com/en-us/research/publication/machine-teaching-new-paradigm-building-machine-learning-systems/
[80] Patrice Y Simard, Saleema Amershi, David M Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, et al. 2017. Machine teaching: A new paradigm for building machine learning systems. *arXiv preprint arXiv:1707.06742* (2017).
[81] Otilia Stretcu, Edward Vendrow, Kenji Hata, Krishnamurthy Viswanathan, Vittorio Ferrari, Sasan Tavakkol, Wenlei Zhou, Aditya Avinash, Enming Luo, Neil Gordon Alldrin, et al. 2023. Agile Modeling: Image Classification with Domain Experts in the Loop. *arXiv preprint arXiv:2302.12948* (2023).
[82] 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 Trans. Interact. Intell. Syst.* 10, 1, Article 7 (aug 2019), 38 pages. https://doi.org/10.1145/3241379
[83] Ben Swanson, Kory Mathewson, Ben Pietrzak, Sherol Chen, and Monica Dinalescu. 2021. Story centaur: Large language model few shot learning as a creative writing tool. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. 244–256.
[84] Saeid Asgari Taghanaki, Aliasghar Khani, Fereshte Khani, Ali Gholami, Linh Tran, Ali Mahdavi-Amiri, and Ghassan Hamarneh. 2022. $MaskTune$: Mitigating Spurious Correlations by Forcing to Explore. *arXiv preprint arXiv:2210.00055* (2022).
[85] Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton van den Hengel. 2022. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16761–16772.
[86] Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, et al. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. *arXiv preprint arXiv:2008.05122* (2020).
[87] 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).
[88] Joshua Vendrow, Saachi Jain, Logan Engstrom, and Aleksander Madry. 2023. Dataset interfaces: Diagnosing model failures using controllable counterfactual generation. *arXiv preprint arXiv:2302.07865* (2023).
[89] James Wexler, Mahima Pushkarna, Tolga Bolukbasi, Martin Wattenberg, Fernanda Viégas, and Jimbo Wilson. 2020. The What-If Tool: Interactive Probing of Machine Learning Models. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 56–65. https://doi.org/10.1109/TVCG.2019.2934619
[90] Olivia Wiles, Isabela Albuquerque, and Sven Gowal. 2022. Discovering bugs in vision models using off-the-shelf image generation and captioning. *arXiv preprint arXiv:2208.08831* (2022).
[91] Tongshuang Wu, Ellen Jiang, Aaron Donsbach, Jeff Gray, Alejandra Molina, Michael Terry, and Carrie J Cai. 2022. PromptChainer: Chaining Large Language Model Prompts through Visual Programming. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 359, 10 pages. https://doi.org/10.1145/3491101.3519729
[92] 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).
[93] Chenyang Yang, Rishabh Rustogi, Rachel Brower-Sinning, Grace A Lewis, Christian Kästner, and Tongshuang Wu. 2023. Beyond Testers' Biases: Guiding Model Testing with Knowledge Bases using LLMs. *arXiv preprint arXiv:2310.09668* (2023).
[94] Huihan Yao, Ying Chen, Qinyuan Ye, Xisen Jin, and Xiang Ren. 2021. Refining language models with compositional explanations. *Advances in Neural Information Processing Systems* 34 (2021), 8954–8967.
[95] Michael Zhang and Christopher Ré. 2022. Contrastive adapters for foundation model group robustness. *Advances in Neural Information Processing Systems* 35 (2022), 21682–21697.
[96] Yuhui Zhang, Jeff Z HaoChen, Shih-Cheng Huang, Kuan-Chieh Wang, James Zou, and Serena Yeung. 2023. Diagnosing and rectifying vision models using language. *arXiv preprint arXiv:2302.04269* (2023).
[97] Xiaojin Zhu. 2015. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.
[98] Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N Rafferty. 2018. An overview of machine teaching. *arXiv preprint arXiv:1801.05927* (2018).

A ADDITIONAL INTERFACE DETAILS

In this section, we provide additional details about the Clarify interface, which we found helpful for eliciting natural language feedback from non-expert users.

Error score. The provided Error Score is a rough proxy for how well a given text description predicts model errors. We emphasize that this score is *not* used in the training process and is only meant to give non-expert users a rough idea of what descriptions are useful. It is computed as follows. Consider input text prompt T, and let D_{correct} and D_{error} be subsets of the validation dataset for a given class that the model made correct and incorrect predictions on, respectively. We denote the cosine similarities between the T and the images in each subset as $S_{\text{correct}} = \{\text{sim}(I, T) \mid I \in D_{\text{correct}}\}$ and $S_{\text{error}} = \{\text{sim}(I, T) \mid I \in D_{\text{error}}\}$. To quantify how well image similarity with T can predict model errors, we compute the best class-balanced binary classification accuracy among similarity thresholds τ. Denoting this accuracy as Acc_τ, the error score is computed as $2 \times (\text{Acc}_\tau - 0.5)$, so that uninformative prompts receive a score of 0 and prompts that perfectly predict model errors receive a score of 1.

Similarity threshold. For each natural language threshold, we determine a similarity threshold τ, which can be chosen by the user after inspecting the similarity scores for a representative sample of images or can be automatically chosen as the threshold that maximizes the Error Score. For each class, only the textual feedback with the highest Error Score is used for retraining. Together with this threshold, we can specify a spurious correlation using a tuple of the form (class label, text prompt, similarity threshold) corresponding to a binary classifier that predicts model errors on that class.

Additional backend features for large datasets. We found that a few more optional features can help annotate spurious correlations in larger datasets like ImageNet. We begin by narrowing down the 1000 classes to a smaller number of classes (e.g., 100) most likely to have identifiable spurious correlations. To do so, we first prune out classes with too low or too high accuracy (e.g. accuracy below 0.2 or above 0.8), to ensure a sufficient number of correct and incorrect predictions for each class. For the remaining classes, we caption each image with an image captioning model [48, BLIP] and use a keyword extraction model [30, KeyBERT] to suggest a pool of up to 50 keywords for each class, a procedure inspired by Kim et al. [38]. Through Clarify, we interact with the top 100 classes according to the maximum error score across the candidate keywords. The user is shown the top 10 candidate keywords during interactions as a helpful starting point. We expect that these features will similarly be helpful for other large datasets.

B EXPERIMENTAL DETAILS

Datasets. We run experiments on three datasets: Waterbirds [72], CelebA [51], and ImageNet [14]. Waterbirds and CelebA have a known spurious correlation between the class label and a spurious attribute; we have access to ground truth spurious attribute labels for these datasets. We use these datasets to evaluate whether Clarify can correct model failures due to spurious correlations. To our knowledge, ImageNet does not have any previously known spurious correlations.

Backbone models. All experiments use pre-trained CLIP models [67] as the feature extractor. The Clarify interface uses the CLIP ViT-L/14 vision and language backbones for calculating image-text similarity. We use the CLIP ResNet-50 and ViT-L/14 models for Waterbirds and CelebA and only the CLIP ViT-L/14 model for ImageNet. We use frozen backbone models and only train a final linear layer for classification, following related works for addressing spurious correlations [39, 95]. We use no data augmentation and normalize all embeddings before computing similarity or training.

Methods. Table 3 and Table 6 show results for Clarify and several representative prior methods for addressing spurious correlations. We experiment with several variants of standard ERM training with a labeled training set: uniform weighting, class-balanced weighting, and "worst-class", a DRO-like weighting scheme that adaptively trains on only the class with the highest loss. We experiment with two variants of training a model with Clarify annotations: reweighting data so that each of the two slices has equal weight (slice-balanced), and a DRO-like weighting scheme which adaptively trains on only the slice with the highest loss (worst-slice).

Annotators. We recruit 26 non-expert users through Prolific (https://www.prolific.co/). These participants had no qualifications beyond being native English speakers and having some programming experience and did not necessarily have any prior knowledge about machine learning. We provide a brief tutorial on using the interface and ask each participant to annotate the class with the highest error rate for each dataset. After completing the user study, we retrained the models for both datasets using each user-provided annotation. The authors of this paper collected another set of annotations for Waterbirds and CelebA, which we use as a baseline for comparison. Additionally, annotations for the ImageNet dataset were collected by paper authors.

C ADDITIONAL DETAILS FOR BIAS-TO-TEXT EXPERIMENT

Here, we provide additional details for the comparison between Clarify and Bias-to-Text [38], an automated bias discovery method. The automated pipeline of Bias-to-Text consists of two steps: (1) extracting keywords from image captions of incorrect examples and (2) ranking these potential keywords based on how well they separate correct and incorrect examples. More specifically, they look for keywords that maximize CLIP score, which is defined as

$$s_{\text{CLIP}}(a; \mathcal{D}) := \text{sim}\left(a, \mathcal{D}_{\text{wrong}}\right) - \text{sim}\left(a, \mathcal{D}_{\text{correct}}\right) \tag{3}$$

where $\mathcal{D}_{\text{wrong}}$ and $\mathcal{D}_{\text{correct}}$ are the sets of incorrect and correct examples, respectively. A keyword with a high CLIP score will likely describe something in common between the incorrect examples and thus may correspond to a spurious correlation. For each keyword, they also report the subgroup accuracy, which is the model's accuracy on the subset of examples containing the keyword. This method is representative of the state-of-the-art in automated bias discovery and was shown to outperform other recent automated bias discovery methods such as ERM confidence [50], Failure Direction [34], and Domino [21].

We evaluated the automated pipeline of Bias-to-Text on several classes in the ImageNet validation set in which we identified spurious correlations and found specific pitfalls that make it difficult to

Spotlight class (ours: "shining")

Keyword	CLIP Score	Subgroup Acc
street lamp	3.32	0.0 (N=1)
lamp	2.50	66.7 (N=6)
top	2.46	0.0 (N=1)
kitchen	2.22	50.0 (N=2)
street	2.07	33.3 (N=3)
suite	1.94	0.0 (N=1)
city	1.87	0.0 (N=3)
room	1.77	0.0 (N=2)
light	1.51	81.8 (N=22)
night	1.12	80.0 (N=5)

Rifle class (ours: "wooden barrel")

Keyword	CLIP Score	Subgroup Acc
person	1.59	14.3 (N=14)
soldier	0.60	0.0 (N=8)
project picture	0.20	0.0 (N=3)
soldiers	0.18	0.0 (N=5)
dark room	-0.47	0.0 (N=1)
machine	-0.95	20.0 (N=5)
gun	-1.69	30.0 (N=20)
nice gun	-1.98	0.0 (N=2)
machine gun	-2.00	20.0 (N=5)
weapons	-2.01	42.9 (N=7)

Academic Gown class (ours: "many people in robes")

Keyword	CLIP Score	Subgroup Acc
person	0.51	25.0 (N=24)
photo	0.50	27.3 (N=11)
graduate	0.23	13.3 (N=15)
graduates	0.21	25.0 (N=8)
graduation	0.16	13.3 (N=15)
pose	0.12	20.0 (N=10)
poses	0.01	0.0 (N=4)
students	-0.25	0.0 (N=7)
graduation ceremony	-0.57	16.7 (N=12)
ceremony	-1.27	23.1 (N=13)

Bighorn class (ours: "rocky hillside")

Keyword	CLIP Score	Subgroup Acc
goat	1.39	5.9 (N=17)
sheep	1.35	0.0 (N=9)
mountain goat	0.11	0.0 (N=2)
biological	0.06	28.6 (N=7)
biological species	0.00	28.6 (N=7)
species	-0.03	28.6 (N=7)
bighorn sheep	-0.04	0.0 (N=4)
bighorn sheep stands	-0.29	0.0 (N=2)
stands	-1.32	0.0 (N=7)
herd	-2.27	14.3 (N=7)

Loupe class (ours: "person holding a magnifying glass")

Keyword	CLIP Score	Subgroup Acc
black	0.01	0.0 (N=4)
camera	-0.08	28.6 (N=7)
book	-0.65	33.3 (N=3)
compact	-0.72	0.0 (N=2)
compact camera	-1.12	0.0 (N=2)
watch	-1.50	0.0 (N=1)
pocket	-1.84	0.0 (N=1)
pocket watch	-2.05	0.0 (N=1)
glass	-2.30	57.1 (N=14)
magnifying glass	-6.19	71.4 (N=7)

Weasel class (ours: "snow weasel")

Keyword	CLIP Score	Subgroup Acc
bear cub sits	1.72	0.0 (N=1)
black bear cub	1.62	50.0 (N=2)
young black bear	0.81	0.0 (N=1)
biological species	-0.17	85.7 (N=14)
dead squirrels	-0.19	0.0 (N=1)
file photo	-0.19	0.0 (N=2)
undated file	-0.36	0.0 (N=2)
undated file photo	-0.41	0.0 (N=2)
grass	-0.87	85.7 (N=7)
squirrels were found	-0.87	0.0 (N=1)

Table 5: Comparison to Bias-to-Text [38], an automated bias discovery method on ImageNet. We show the top 10 keywords identified by Bias-to-Text in descending order of their recommended score. We also show the text feedback provided through Clarify for comparison. The keywords identified by Bias-to-Text often include irrelevant words or correspond to very small subpopulations, indicating that current automated methods ultimately require human oversight or intervention to discover the most relevant and biased subpopulations.

use alone in practice. In Table 5, we show 10 keywords identified by Bias-to-Text for four of the classes for which we identified spurious correlations. We note that the top identified keywords, i.e., the ones with the highest CLIP score, often describe something highly related to the class label, such as "goat" for the "bighorn". We further note that the method also identifies very small subpopulations, for example "baby shower" which only appears in three of the 50 examples in the "bakery" class. Text feedback from Clarify never the top keyword recommended by Bias-to-Text, and was only in the top 10 for 5 out of 31 classes.

In its current form, automated bias discovery methods such as Bias-to-Text ultimately require oversight to identify the most relevant keywords. The human-in-the-loop nature of Clarify can be seen as recognizing this dependency and providing a more direct way for users to inspect and correct model failures. However, we note that automated discovery methods are still highly useful in the context of Clarify, as they can prime annotators with a set

of candidate keywords or help prioritize the most promising parts of the dataset. We believe a more integrated combination of automated discovery methods and human-in-the-loop methods such as Clarify will be a fruitful direction for future work.

Finally, we compare the annotations discovered by Clarify and Bias-to-Text. We take the top keyword identified by Bias-to-Text for each class and compare the model's accuracy on the majority and minority splits in Figure 9. The annotations for Bias-to-Text show substantially higher minority split accuracy (Clarify 21.1%, Bias-to-Text 45.2%), with a smaller gap with the majority split. Furthermore, after re-training with these annotations using our reweighting procedure, we observed a slight decrease in held-out minority split accuracy (45.2% to 44.3%). This is in contrast to re-training with Clarify annotations, which substantially improved minority split accuracy (21.1% to 28.7%). These results indicate that automated bias discovery methods such as Bias-to-Text fail to identify the most relevant or consistent subpopulations, highlighting the need for oversight.

D QUALITATIVE FEEDBACK

We collected qualitative feedback from the non-expert participants in our user study. We found that the interface was easy to use and provided useful insights into the strengths and weaknesses of the method in Table 7.

	Assumptions	Method	Waterbirds WG	Waterbirds Avg	Waterbirds Gap	CelebA WG	CelebA Avg	CelebA Gap
CLIP ResNet-50	Zero-Shot	Class Prompt	36.6	92.2	55.6	74.0	81.9	7.9
		Group Prompt	55.9	87.8	31.9	70.8	82.6	11.8
	Labels	ERM	7.9	93.5	85.6	11.9	94.7	82.8
		ERM (ours)	63.4	96.0	32.6	31.1	95.4	64.3
		ERM (ours, class-balanced)	48.6	95.2	46.7	65.8	93.4	27.6
		ERM (ours, worst-class)	55.9	95.8	39.9	56.9	94.1	37.2
	Labels, Text Feedback	Clarify (avg non-expert)	69.8	84.1	13.3	83.7	93.2	9.5
		Clarify (best non-expert)	**82.5**	90.7	8.2	88.8	92.9	4.1
		Clarify (author)	75.7	83.8	**8.1**	**89.1**	92.1	**3.0**
	Labels, Group Annotation	DFR (downsample)	63.9	91.8	27.9	76.9	92.5	15.6
		DFR (upsample)	51.3	92.4	41.1	89.6	91.8	2.2
		DFR (our implementation)	78.7	90.8	12.1	**90.6**	91.9	**1.3**
		Group DRO (our implementation)	**81.3**	88.1	**6.8**	89.2	91.8	2.7
	Labels, Additional Params	ERM Adapter	60.8	96.0	35.2	36.1	94.2	58.1
		WiSE-FT	49.8	91.0	41.2	85.6	88.6	3.0
		Contrastive Adapter	**83.7**	89.4	**5.7**	**90.0**	90.7	**0.7**
CLIP ViT-L/14	Zero-Shot	Class Prompt	25.7	87.3	61.6	62.1	71.9	9.8
		Group Prompt	27.4	85.5	58.1	72.4	81.8	9.4
	Labels	ERM	65.9	97.6	31.7	28.3	94.7	66.4
		ERM (our implementation)	79.5	97.4	17.9	25.7	94.6	68.9
		ERM (our implementation, class-balanced)	71.1	97.2	26.1	63.7	92.6	28.9
		ERM (our implementation, worst-class)	74.3	97.1	22.8	56.9	93.3	36.4
	Labels, Text Feedback	Clarify (author)	**81.8**	96.8	**14.9**	**88.8**	90.9	**2.1**
	Labels, Group Annotation	DFR (downsample)	51.9	95.7	43.8	76.3	92.1	15.8
		DFR (upsample)	65.9	96.1	30.2	83.7	91.2	7.5
		DFR (our implementation)	85.9	93.5	7.6	**89.0**	90.9	**1.9**
		Group DRO (our implementation)	**88.5**	92.7	**4.1**	88.1	91.1	2.9
	Labels, Additional Params	ERM Adapter	78.4	97.8	19.4	36.7	94.2	57.5
		WiSE-FT	65.9	97.6	31.7	80.0	87.4	7.4
		Contrastive Adapter	**86.9**	96.2	**9.3**	**84.6**	90.4	**5.8**

Table 6: Evaluation of methods for improving group robustness of CLIP models. Grouped by data and expressivity, with best worst-group (WG) and robustness gaps bolded. All metrics are averaged over three seeds.

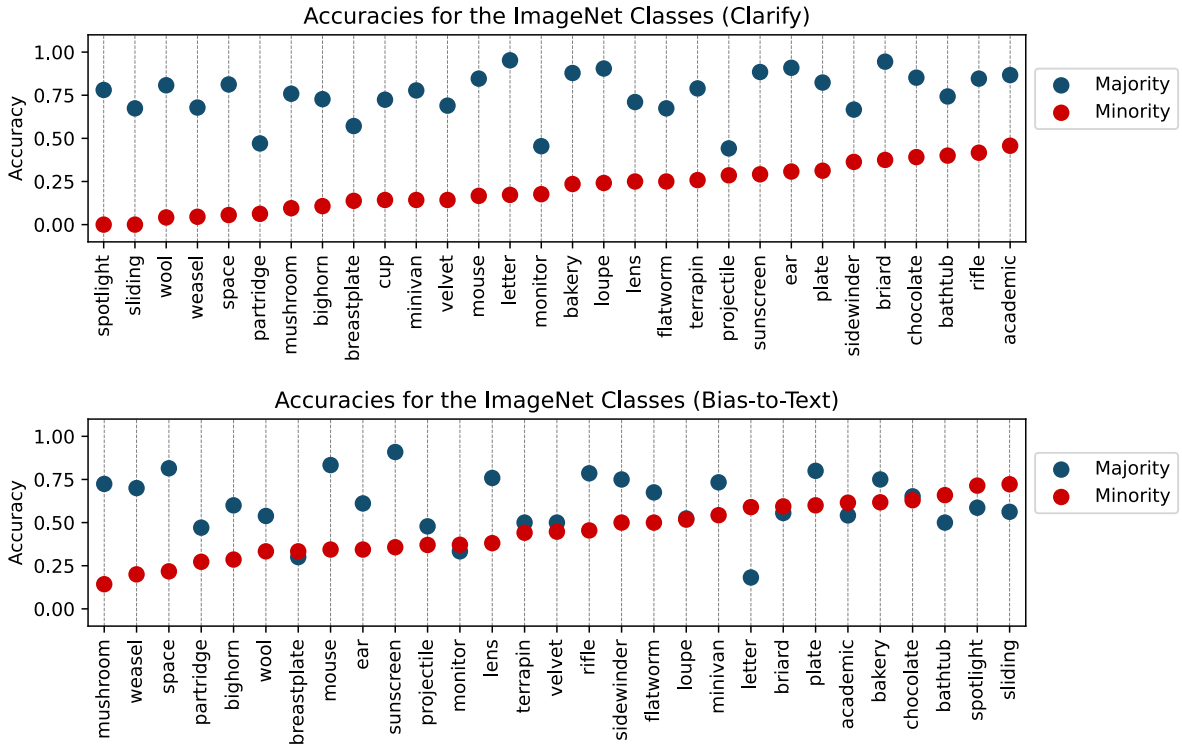

Figure 9: Comparison of annotations discovered by Clarify and Bias-to-Text.We show the accuracy of an ImageNet-trained model on the validation setFor each class with an identified spurious correlation, we show majority split and minority split accuracy. The annotations for Bias-to-Text show substantially higher minority split accuracy (Clarify: 21.1%, Bias-to-Text 45.2%), with a smaller gap with the majority split.This indicates that Clarify was substantially more accurate in identifying hard subpopulations.

Clarity
> Study seemed a little confusing
> It was a little confusing at first, but then became clear and challenging.
> Compared to the example task, they were a bit more difficult but still understandable.
> Yes I understood what was being measured.
Strategies and Thought Processes
> Yes, to find a pattern among the red (incorrect) squares and determine a relevant phrase or description that captures the similarity
> Yes. I just tried to figure out what characteristics mainly led to something wrong.
> I was honestly having fun trying different prompts.
> It was clear in that I had to try to figure out the AI's weakness, but finding that weakness was hard. I tried to do related topics, then used what I saw in the pictures for fodder.
> Yes, I would look for what was different about the incorrect images and enter my first guess then work from there.
> I was trying to spot what the common things were that the AI was struggling to pick up in the photos it was getting wrong.
> The study seemed to be about helping correct the misbehavior of AI. My thought process was mostly linked to trying to find shared features that miscategorized images included.
> The only thing I wasn't really sure of was how detailed we could be, how many criteria we could give. I tried to keep it low (one of my higher scorers was just m̈en̈) but sometimes I had to chain them (d̈ark backgrounds and a lot of treesör similar) to score well.
Difficulty
> The last one was pretty difficult, but I think I saw all the images correctly. For a second I thought one was a young blonde Trudeau with a flipped-up haircut, but realized that wasn't him
> more difficult than I thought to come up with various prompts
Suggestions for Improvement
> It might be nice if our most recent guess was highlighted in the table on the left. Every time I'd submit a guess I found my self trying to remember exactly how I'd phrased it and trying to find it in the table to see how well I did.
> Look at the different images, try to find commonalities that might be affecting identification

Table 7: Open-ended qualitative feedback from participants, grouped by topic.

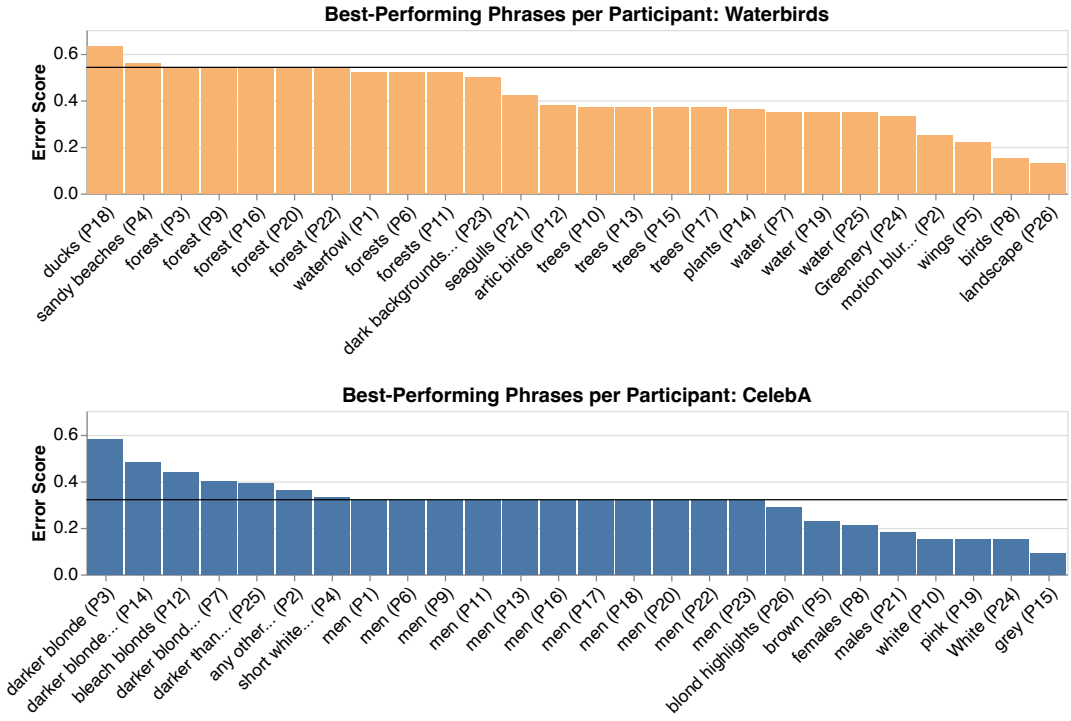

Figure 10: Non-experts used Clarify to identify high-quality descriptions with Error Scores that matched or exceeded the authors' expert annotations.

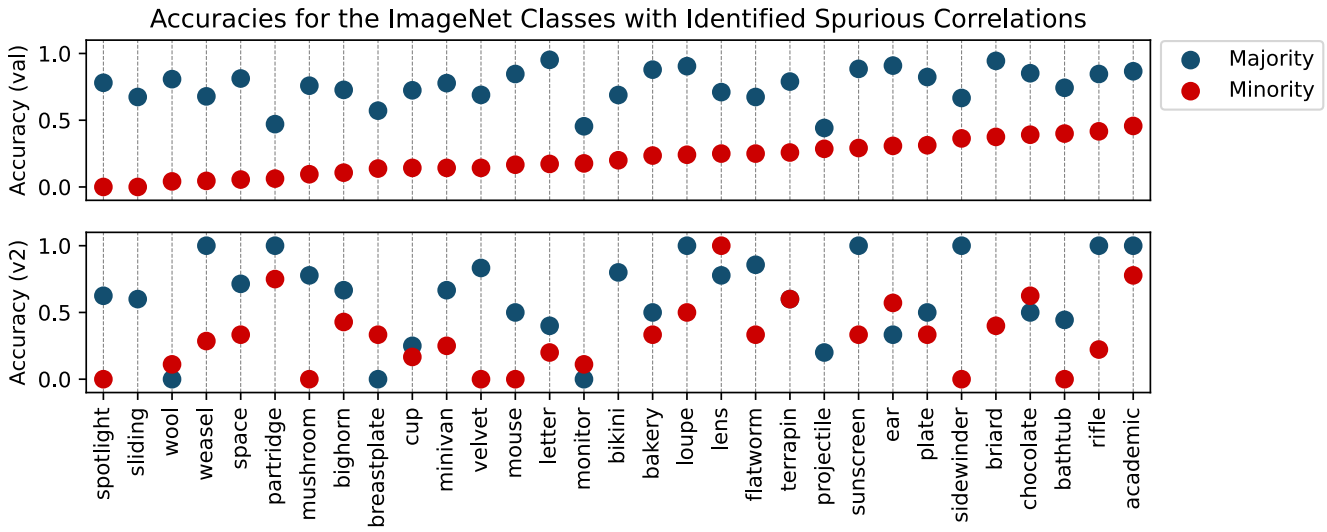

Figure 11: Accuracy of a model trained on the ImageNet train set, on the ImageNet validation set (top), and on ImageNet-V2 (bottom). For each class with an identified spurious correlation, we show majority and minority split accuracy. The model achieves lower accuracy on the minority split for all classes in the validation set and all but 6 classes in ImageNet-V2, indicating that the model relies on each identified spurious feature.