Where Does My Model Underperform? A Human Evaluation of Slice Discovery Algorithms

Nari Johnson [1] **Ángel Alexander Cabrera** [1] **Gregory Plumb**[† 1 2] **Ameet Talwalkar** [1]

Abstract

Machine learning (ML) models that achieve high average accuracy can still underperform on semantically coherent subsets ("slices") of data. This behavior can have significant societal consequences for the safety or bias of the model in deployment, but identifying these underperforming slices can be difficult in practice, especially in domains where practitioners lack access to group annotations to define coherent subsets of their data. Motivated by these challenges, ML researchers have developed new *slice discovery algorithms* that aim to group together coherent and high-error subsets of data. However, there has been little evaluation focused on whether these tools help humans form correct hypotheses about where (for which groups) their model underperforms. We conduct a controlled user study ($N = 15$) where we show 40 slices output by two state-of-the-art slice discovery algorithms to users, and ask them to form hypotheses about an object detection model. Our results provide positive evidence that these tools provide benefit over a naive baseline, and challenge dominant assumptions shared by past work.

1. Introduction

A growing number of works propose tools to help stakeholders form hypotheses about the behavior of machine learning (ML) models. One type of behavior that can have significant societal consequences occurs when a model underperforms on semantically coherent subsets (*i.e.,* "slices") of data. For example, Buolamwini & Gebru (2018) found that leading tech companies' commercial facial recognition models were significantly less accurate at classifying faces of women with darker skin. Knowledge of this model behavior informed advocacy efforts that led to fundamental changes to dataset curation for facial recognition (IBM, 2018; Birhane, 2022) and public deliberation surrounding governance of facial recognition systems (Raji & Buolamwini, 2022). More broadly, knowledge of underperforming slices can inform model selection and deployment (Balayn et al., 2023) or actions that can be taken to fix the model (Holstein et al., 2019; Cabrera et al., 2021; Idrissi et al., 2022).

However, identifying these underperforming slices can be difficult in practice. In many domains, practitioners often do not have access to group annotations that can be used to define semantically *coherent* (*i.e.,* united by a single human-understandable concept) subsets of their data (Cabrera et al., 2023b). Motivated by these challenges, ML researchers have developed new automated tools in the growing field of *slice discovery*. At a high level, these *slice discovery algorithms* are unsupervised methods that aim to group together coherent and high-error slices of data (Sohoni et al., 2020; d'Eon et al., 2021; Singla et al., 2021; Eyuboglu et al., 2022; Plumb et al., 2023; Wang et al., 2023). These works propose that a human stakeholder can inspect the slices output by these algorithms to form hypotheses of model behavior, *i.e.,* describe in words a subgroup where the model underperforms.

While researchers continue to develop new slice discovery algorithms, there has been little evaluation of whether these algorithms help stakeholders achieve their proposed goals. Past human evaluations of slice discovery tools have used subjective judgments of the output slices' coherence (such as whether users can find a description that "matches" the majority of images in the slice) as proxies of these algorithms' utility (Singla et al., 2021; d'Eon et al., 2021), but stop short of verifying whether these descriptions are useful or even accurate depictions of the model's behavior. In this work, we ask: Do the slices output by these algorithms help users form *correct hypotheses of model behavior?*

As a motivating example, consider a scenario where a practitioner wishes to evaluate a new object detection model designed to be deployed in an autonomous vehicle. The practitioner could run a slice discovery algorithm on a dataset of dash-cam photos collected from field testing, but its output (groups of photos) is not immediately actionable. De-

† The author completed the majority of the work as a student at CMU. [1]Carnegie Mellon University [2]Inpleo, Inc. Correspondence to: Nari Johnson <narij@andrew.cmu.edu>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Honolulu, Hawaii, USA. Copyright 2023 by the author(s).

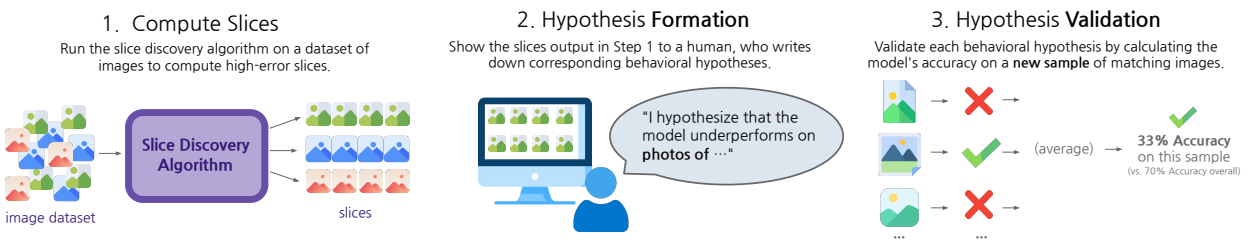

Figure 1. An overview of the study in three steps. (Left) We run different slice discovery algorithms to compute high-error slices (subsets of the input dataset). (Middle) We conduct a human subject study where we *show the slices from Step 1* to a human subject, who forms a hypothesis (i.e. description of a subgroup where the model underperforms) corresponding to each slice. (Right) We validate each user-generated hypothesis from Step 2 by calculating the model's accuracy on a new sample of images that match the user's description.

scribing in words a group where the model underperforms (*e.g.,* "the model fails to identify stop signs in snowy environments") is a prerequisite step for several downstream actions that the practitioner's company could take, such as initiating targeted data collection efforts (*e.g.,* additional field testing to gather more data in snowy regions) or selective deployment (*e.g.,* delaying roll-out of the new model in areas that experience snow).

Unfortunately, taking action to address an *incorrect* hypothesis can have several undesirable consequences. For example, if the model actually performs just as well in snowy environments, then the company's efforts to improve the model's performance on that group may have been better expended elsewhere, or may even worsen the model's performance on other groups (Li et al., 2022). Thus, we argue that this under-examined step of hypothesis formation is critically important for many stakeholders.

Our Contributions. We design a controlled user study to investigate how participants make sense of the slices output by algorithms to form hypotheses of model behavior. Our study has three parts, summarized in Figure 1. Our work *benchmarks* two state-of-the-art slice discovery tools, characterizes how users may *(mis)interpret* existing tools, and identifies *design opportunities* for ML and HCI researchers. In the following sections, we provide an overview of slice discovery, motivate our research hypotheses and study design, and present results from our human subject evaluation. A full version of this paper is available online at: https://arxiv.org/abs/2306.08167

2. Slice Discovery Preliminaries

We focus our study on *slice discovery algorithms*: fully automated methods that aim to partition the data into coherent and high-error subsets (Eyuboglu et al., 2022; Plumb et al., 2023). We exclude methods that rely on additional information such as fine-grained group annotations (Polyzotis et al., 2019; Liu et al., 2021) or human supervision to define coherent subsets. Consequently, the methods we study do *not* require the practitioner to anticipate the types of inputs where the model may underperform in advance. We discuss related work on past evaluations of slice discovery algorithms in Appendix A.

Definition We follow Plumb et al. (2023) and define a slice discovery algorithm as a method that given as input a trained model f and dataset of labeled images $D = \{(x_i, y_i)\}_{i=1}^{n}$[1], outputs k slices $[\Psi_j]_{j=1}^{k}$. Each slice $\Psi_j \subseteq D$ is a subset of the input data. The objective of slice discovery is for each slice Ψ_j to correspond to a single coherent subgroup where the model indeed underperforms (i.e. f has lower accuracy for images in Ψ_j than for all images in D). Past works propose that a practitioner can manually inspect the datapoints belonging to each slice to "*identify common attributes*" (Eyuboglu et al., 2022) to describe the underperforming subgroups.

Methods We evaluate two slice discovery algorithms: DOMINO (Eyuboglu et al., 2022) and PLANESPOT (Plumb et al., 2023). Both algorithms work by running error-aware clustering on an embedding of each image, and differ in the specific clustering algorithm and embedding that they use. We describe each algorithm in detail in Appendix B. We chose these two algorithms because they achieved state-of-the-art performance on past benchmarks (Eyuboglu et al., 2022; Plumb et al., 2023) at the time we ran our study.

Our BASELINE algorithm randomly samples from the set of misclassified images, without replacement. While a subset of random misclassified images is high-error, it is unlikely to be coherent. Our baseline was designed to simulate a naive workflow where a practitioner inspects an unsorted sample of misclassified images to form hypotheses.

[1]We follow past work and give a separate held-out test set that the model f was *not* trained on as the input to a slice discovery algorithm.

Validating User Hypotheses To evaluate each slice discovery algorithm, our study focuses on the human-centered task of *hypothesis formation*. We define a "*hypothesis*" as a *user-generated text description* of a group where they believe that the model underperforms. Intuitively, a "correct hypothesis" should describe a group where the model has significantly lower accuracy. If Φ denotes the set of all images in D that belong to the group, we define the *performance gap* $Gap(f, \Phi)$ of f on Φ as

$$\underbrace{\frac{1}{|D|}\sum_{(x,y)\in D} 1(f(x)=y)}_{\text{average accuracy on } D} - \underbrace{\frac{1}{|\Phi|}\sum_{(x,y)\in \Phi} 1(f(x)=y)}_{\text{average accuracy on } \Phi \subset D} \quad (1)$$

We define the correctness of the hypothesis "f underperforms on Φ" by thresholding the gap: $Gap(f, \Phi) \geq \tau$, where threshold $\tau \geq 0$ is a hyperparameter that defines the minimum performance gap necessary for a hypothesis to be "correct". For example, if we set $\tau = 0.2$, then a "correct" hypothesis describes a group where the model has 20% worse accuracy on images that belong to the group, compared to all images that belong to its class.

However, in many settings, we do not have access to Φ, the complete set of images in D that match each hypothesis. Therefore, we decided to *approximate* the model's performance gap for each hypothesis using a sample $\hat{\Phi}$ of images that match each hypothesis, where $\hat{\Phi} \subset \Phi$. To retrieve $\hat{\Phi}$, we followed past work (Gao et al., 2022) and ranked candidate images using their CLIP similarity score (Radford et al., 2021) to the hypothesis text description. We then manually annotated whether each retrieved image matched the user's hypothesis. We provide an extended description of our approximation strategy and discuss several ablations in Appendix C.

3. Experimental Design

We present an overview of our research questions and study design, and provide complete details and screenshots of the study interface in Appendix E.

Our primary research hypotheses aim to benchmark two state-of-the-art slice discovery tools ("conditions"), DOMINO and PLANESPOT, against a naive baseline. This comparison provides an important sanity check that these tools may offer users some benefit over a naive workflow of inspecting a random sample of errors.

First, we hypothesize that users' hypotheses corresponding to the slices output by algorithms designed for slice discovery are more likely to be "correct":

H1. A greater proportion of users' hypotheses corresponding to slices output by slice discovery algorithms will be correct, when compared to hypotheses corresponding to slices output by the naive baseline.

In addition to validating users' hypotheses, we also asked them to self-report how *difficult* it was to form their hypothesis (*i.e.,* describe the slice):

H2. Users will rate slices output by slice discovery algorithms as *easier to describe*, when compared to slices output by the naive baseline.

We also asked users to select all of the images that *match their hypothesis* from a sample of the top-20 images that belong to the slice. The number of matching images is one measure of the *coherence* of the slice (*i.e.,* the extent to which all of the images share a common description):

H3. Users will *select more images as "matching" their hypothesis* for slices output by slice discovery algorithms, when compared to slices output by the naive baseline.

Lastly, our final hypothesis studies whether coherence (when many images in the slice share a common description) implies correctness (that the model does in fact underperform on *all* images that match this description), a common assumption shared by past work (see Appendix A):

H4. The number of images in the slice that match a user's hypothesis (a measure of slice coherence) does not predict whether their hypothesis is correct.

To conduct our study, we recruited 15 participants who are current students that had self-reported "intermediate knowledge in machine learning (ML) or computer vision (CV)". For the model f, we fine-tuned a pre-trained ResNet-18 on data from MS-COCO (Lin et al., 2014), a large-scale object detection task that we chose for its accessibility to a wide audience of non-expert stakeholders.

We generated 60 total slices (20 slices per each of the three algorithms described in Section 2) to show users. Rather than running each slice discovery algorithm on the entire COCO test set (which contains 91 object types), we ran each algorithm on subsets of the data that all have the same object (*e.g.,* we gave as input all images that have a ground-truth label of `airplane`). We used each slice discovery algorithm to return the top $k = 4$ slices for 5 different objects, chosen randomly from a list of candidate objects where the model had at least 50% recall: `airplane`, `train`, `giraffe`, `skis`, and `broccoli`.

Study Procedure. We began each study by presenting the user with a description of the study task and walk-through of the study interface. Then, we asked users to complete a *questionnaire* where they were asked to write down a behavioral hypothesis (*i.e.,* describe in words a group that matches as many images in the slice as possible) for 12 different slices. Users were also asked to select all images in the slice that matched their hypothesis and rate how difficult it was

for them to describe the slice. We detail how we present each slice and share the complete study questionnaire in Appendix E.

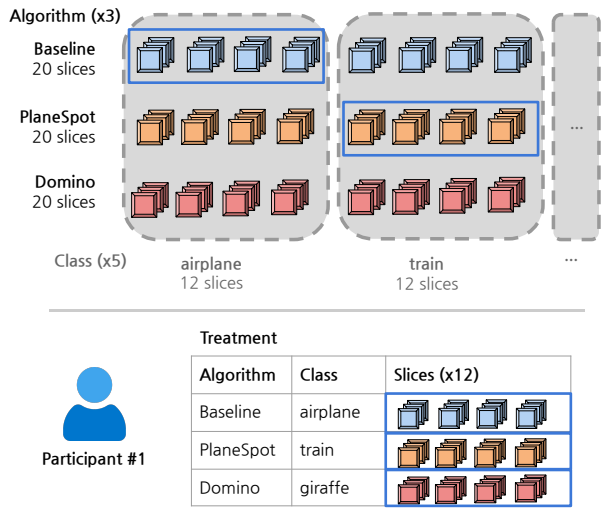

Figure 2. **Experimental Setup.** (Top) We collect users' hypotheses for 60 total slices. For each of the 3 algorithms (rows), and for each of the 5 classes (columns), we compute the top-4 slices. We show 2 out of 5 classes (and 24 out of 60 total slices) in the figure due to space constraints. (Bottom) Each study participant was shown 12 total slices output by 3 different algorithms, and saw slices corresponding to a different class for each algorithm. The blue boxes on the top panel highlight the slices shown to Participant #1.

Experimental Design. We used a within-subjects design shown in Figure 2, where each participant was randomly assigned to 3 (out of 5 candidate) object classes. All participants were presented with 12 total slices, and saw slices output by all 3 algorithms. The 3 algorithm conditions were presented to participants in a random order to control for learning effects. We showed each of the 60 total slices to 3 different participants, collecting 3 hypotheses per each slice. Users were blinded to the study condition when completing each questionnaire: while they were informed that each slice was computed by an algorithm, they were not given any additional details about *which algorithm* was used to compute each slice.

4. Results

To evaluate **H1** (whether the average number of correct hypotheses varies across conditions), we exclude hypotheses where we failed to find a sufficiently large sample of matching images (*i.e.,* at least 15) to approximate the performance gap. Applying this criteria, we retain 135 out of the original 180 hypotheses (75%). To evaluate **H2** and **H3**, we calculate each measure using all 180 of the original hypotheses.

To test for statistically significant differences between the three algorithm conditions, we ran ANOVA tests with Tukey post-hoc tests for multiple comparisons to compare the proportion of correct hypotheses (**H1**) and number of matching images (**H3**). We ran Mann-Whitey tests with Bonferroni corrections to compare the Likert-scale self-reported difficulty of describing each slice (**H2**). All measures were are stratified by algorithm condition and aggregated across all users. We used a p-value of 0.05 as our cutoff for significance. We present additional details for each statistical test (*e.g.,* test statistics) and additional analyses in Appendix J.

Correctness. For each condition, we calculated the proportion of hypotheses that are "correct" using a performance gap threshold of at least 20%. We found that 69 out of 135 total hypotheses (51%) are correct (standard error 4.3%). When we stratify by condition, 35% of BASELINE, 49% of PLANESPOT, and 72% of DOMINO hypotheses are correct, with standard errors 6.7%, 7.9%, and 6.9% respectively (Figure 3). We found a statistically significant difference between the BASELINE and DOMINO conditions only ($p < 0.001$), partially supporting **H1**. We present additional results where we ablate the performance gap threshold in Appendix J.1. In summary, we found that a consistent trend (that DOMINO outperforms PLANESPOT, which outperforms BASELINE) holds for all variations tried.

Number of matching images. Overall, we observed that users selected more images as "matching" their hypothesis for slices output by slice discovery algorithms relative to the naive baseline, supporting **H2**. Appendix Figure 9 shows a histogram of the empirical distribution of matching images for each condition. On average, users selected 12.9 and 12.8 out of the 20 displayed images in the slice as "matching" their hypotheses for the DOMINO and PLANESPOT conditions respectively (standard errors 0.57 and 0.71), vs. 8.8 images (standard error 0.60) for the BASELINE hypotheses. We observed significant pairwise differences between the two slice discovery conditions vs. the baseline condition (with $p < 0.0001$ for both DOMINO and PLANESPOT), and no significant difference between the two slice discovery conditions. However, we note that while the *average* number of matching images is higher for both slice discovery conditions, there is still high *variance* across hypotheses. For example, 30% of hypotheses from the DOMINO or PLANESPOT conditions had < 10 matching images.

Self-reported Difficulty. Overall, we observed that users rated slices output by the slice discovery algorithms as *easier to describe* compared to slices output by the naive baseline condition, supporting **H3**. Figure 4 shows the distribution of Likert-scale ratings for each condition. We observed significant pairwise differences between the two slice discovery conditions vs. the baseline ($p < 0.0001$), and no significant difference between the two slice discovery con-

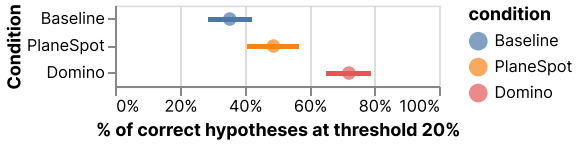

Figure 3. **Hypothesis Correctness.** The percentage of hypotheses per condition that are "correct" using a performance gap threshold $\tau = 20\%$ with standard error bars. Percentages are calculated for the subset of hypotheses that we have a sufficiently large number of examples (*i.e.,* at least 15 matching images) to approximate the performance gap. We find that a greater proportion of users' hypotheses from the PLANESPOT and DOMINO conditions are correct relative to the BASELINE condition.

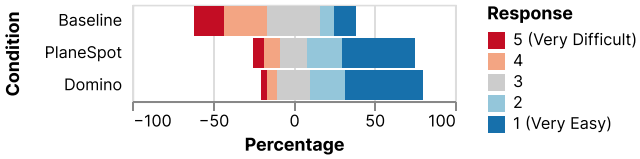

Figure 4. **Self-Reported Difficulty.** A diverging stacked bar chart centered around the netural response of users' self-reported difficulty of describing each slice (on a five-point Likert Scale), stratified by each condition (slicing algorithm). We find that users are significantly more likely to rate slices output by the PLANESPOT and DOMINO conditions as being easy to describe.

ditions.

Does coherence imply correctness? To examine whether coherence (*i.e.,* the number of images in the slice that match a hypothesis) is an appropriate proxy of the hypothesis's correctness (**H4**), we ran two Spearman's rank correlation tests. For both tests, the independent variable is the number of matching images. We tried two dependent variables: the value of the approximate performance gap for the hypothesis (defined in Equation 1), and an indicator for hypothesis correctness using performance gap threshold $\tau = 0.2$. For these analyses only, we retained only the hypotheses that corresponded to slices output by slice discovery algorithms (*i.e.,* slices that were designed to be coherent), and excluded slices output by the baseline condition.

We found *no significant association* between the number of images that match each hypothesis and its correctness, supporting **H4**. For both dependent variables, the number of matching images is only weakly correlated with the hypothesis's correctness, with correlation coefficients $r_s = 0.05$ with $p = 0.6194$ for the value of the performance gap, and $r_s = 0.07$ with $p = 0.5027$ for the correctness indicator. When we examined the hypotheses that had the highest number of matching images (at least 15 out of 20), only 59% of these hypotheses were correct, a rate *lower* than the overall base rate of 61% accuracy for all hypotheses corresponding to slices output by either DOMINO or PLANESPOT.

5. Discussion

Researchers continue to develop new slice discovery tools, but there has been little evaluation of if, and how, humans can make sense of their output. In our controlled study with 15 participants, we found preliminary evidence that existing tools may offer some benefit relative to a naive baseline of examining a random sample of errors. Yet, we found that the slices output by existing tools do not *always* help users form correct hypotheses. For example, only 49% of users' hypotheses corresponding to slices output by PLANESPOT described groups where the model performed at least 20% worse. Thus, existing tools have the potential to mislead users to develop false beliefs if used as proposed by past work.

Our results also challenge several dominant assumptions shared by past work on slice discovery. Several works have implied that if a user can describe all the images in a slice, then the model underperforms on all images that match their description. Our finding that there is no significant association between slice *coherence* and hypothesis *correctness* shows that this assumption is misleading. This finding has important consequences for both evaluation and everyday use of slice discovery tools. We caution researchers away from using the number of images that match a common description as a measure of their algorithms' utility.

Our findings point to several design opportunities for ML researchers and UI designers, which we discuss in detail in Appendix K. One example direction that has been neglected by past work is how the output slices should be visualized and presented to users. Another promising direction is to develop tools that allow users to validate their own hypotheses in real time. Finally, we encourage researchers to re-imagine where and how automation can help users discover underperforming groups. One could imagine more bespoke and interactive workflows that better utilize stakeholders' domain knowledge to define coherent subsets of data, or leverage automation to help stakeholders refine their hypotheses. More broadly, we urge researchers to consider how we can design better algorithmic support for the fundamentally human task of hypothesis formation. We hope that our work is a first step towards centering users when designing and evaluating new tools for slice discovery.

Acknowledgements

We thank our study participants and annotators who made our research possible. We are grateful for helpful feedback from Sherry Tongshuang Wu, Donald Bertucci, Valerie Chen, Vijay Viswanathan, Katelyn Morrison, and Nupoor Gandhi. This work was supported in part by the National Science Foundation grants IIS1705121, IIS1838017, IIS2046613, IIS2112471, and funding from Meta, Morgan Stanley, Amazon, and Google. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of these funding agencies.

References

Balayn, A., Rikalo, N., Yang, J., and Bozzon, A. Faulty or ready? handling failures in deep-learning computer vision models until deployment: A study of practices, challenges, and needs. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3581555. URL https://doi.org/10.1145/3544548.3581555.

Bertucci, D., Hamid, M. M., Anand, Y., Ruangrotsakun, A., Tabatabai, D., Perez, M., and Kahng, M. Dendromap: Visual exploration of large-scale image datasets for machine learning with treemaps, 2022.

Birhane, A. The unseen black faces of ai algorithms, 2022.

Buolamwini, J. and Gebru, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pp. 77–91. PMLR, 2018.

Cabrera, Á. A., Druck, A. J., Hong, J. I., and Perer, A. Discovering and validating ai errors with crowdsourced failure reports. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–22, 2021.

Cabrera, A. A., Fu, E., Bertucci, D., Holstein, K., Talwalkar, A., Hong, J. I., and Perer, A. Zeno: An interactive framework for behavioral evaluation of machine learning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023a. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3581268. URL https://doi.org/10.1145/3544548.3581268.

Cabrera, A. A., Tulio Ribeiro, M., Lee, B., Deline, R., Perer, A., and Drucker, S. M. What did my ai learn? how data scientists make sense of model behavior. *ACM Trans. Comput.-Hum. Interact.*, 30(1), mar 2023b. ISSN 1073-0516. doi: 10.1145/3542921. URL https://doi.org/10.1145/3542921.

d'Eon, G., d'Eon, J., Wright, J. R., and Leyton-Brown, K. The spotlight: A general method for discovering systematic errors in deep learning models. *arXiv preprint arXiv:2107.00758*, 2021.

Eyuboglu, S., Varma, M., Saab, K. K., Delbrouck, J.-B., Lee-Messer, C., Dunnmon, J., Zou, J., and Re, C. Domino: Discovering systematic errors with cross-modal embeddings. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=FPCMqjI0jXN.

Gajos, K. Z. and Chauncey, K. The influence of personality traits and cognitive load on the use of adaptive user interfaces. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, IUI '17, pp. 301–306, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450343480. doi: 10.1145/3025171.3025192. URL https://doi.org/10.1145/3025171.3025192.

Gao, I., Ilharco, G., Lundberg, S., and Ribeiro, M. T. Adaptive testing of computer vision models, 2022.

Gaube, S., Suresh, H., Raue, M., Lermer, E., Koch, T., Hudecek, M., Ackery, A., Grover, S., Coughlin, J., Frey, D., Kitamura, F., Ghassemi, M., and Colak, E. Non-task expert physicians benefit from correct explainable ai advice when reviewing x-rays, 2023.

Holstein, K., Vaughan, J. W., Daumé, H., Dudik, M., and Wallach, H. Improving fairness in machine learning systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, may 2019. doi: 10.1145/3290605.3300830. URL https://doi.org/10.1145%2F3290605.3300830.

IBM. Ibm response to "gender shades: Intersectional accuracy disparities in commercial gender classification". 2018. URL http://gendershades.org/docs/ibm.pdf.

Idrissi, B. Y., Arjovsky, M., Pezeshki, M., and Lopez-Paz, D. Simple data balancing achieves competitive worst-group-accuracy, 2022.

Kumar, A., Raghunathan, A., Jones, R., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution, 2022.

Li, Z., Evtimov, I., Gordo, A., Hazirbas, C., Hassner, T., Ferrer, C. C., Xu, C., and Ibrahim, M. A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others. *arXiv preprint arXiv:2212.04825*, 2022.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021.

Moore, S., Liao, Q. V., and Subramonyam, H. Failurenotes: Supporting designers in understanding the limits of ai models for computer vision tasks. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394215. doi: 10.1145/3544548.3581242. URL https://doi.org/10.1145/3544548.3581242.

Plumb, G., Johnson, N., Ángel Alexander Cabrera, and Talwalkar, A. Towards a more rigorous science of blindspot discovery in image classification models, 2023.

Polyzotis, N., Whang, S., Kraska, T. K., and Chung, Y. Slice finder: Automated data slicing for model validation. In *Proceedings of the IEEE Int' Conf. on Data Engineering (ICDE), 2019*, 2019. URL https://arxiv.org/pdf/1807.06068.pdf.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021.

Rajani, N., Liang, W., Chen, L., Mitchell, M., and Zou, J. Seal : Interactive tool for systematic error analysis and labeling, 2022.

Raji, I. D. and Buolamwini, J. Actionable auditing revisited: Investigating the impact of publicly naming biased performance results of commercial ai products. *Commun. ACM*, 66(1):101–108, dec 2022. ISSN 0001-0782. doi: 10.1145/3571151. URL https://doi.org/10.1145/3571151.

Shankar, V., Roelofs, R., Mania, H., Fang, A., Recht, B., and Schmidt, L. Evaluating machine accuracy on ImageNet. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8634–8644. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/shankar20c.html.

Singla, S., Nushi, B., Shah, S., Kamar, E., and Horvitz, E. Understanding failures of deep networks via robust feature extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12853–12862, 2021.

Sohoni, N., Dunnmon, J., Angus, G., Gu, A., and Ré, C. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

Suresh, H., Shanmugam, D., Bryan, A., Chen, T., D'Amour, A., Guttag, J. V., and Satyanarayan., A. Kaleidoscope: Semantically-grounded, context-specific ml model evaluation. In *CHI Conference on Human Factors in Computing Systems*, 2023.

Vasudevan, V., Caine, B., Gontijo-Lopes, R., Fridovich-Keil, S., and Roelofs, R. When does dough become a bagel? analyzing the remaining mistakes on imagenet, 2022.

Wang, F., Adebayo, J., Tan, S., Garcia-Olano, D., and Kokhlikyan, N. Error discovery by clustering influence embeddings. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023. URL https://openreview.net/forum?id=Rj8TUzl8nT8.

Wu, T., Ribeiro, M. T., Heer, J., and Weld, D. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 747–763, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1073. URL https://aclanthology.org/P19-1073.

Yuksekgonul, M., Bianchi, F., Kalluri, P., Jurafsky, D., and Zou, J. When and why vision-language models behave like bags-of-words, and what to do about it?, 2023.

A. Related Work

Automated Evaluations of Slice Discovery Tools Automated evaluations of slice discovery tools fall into three categories. The first category of evaluations measure the quality of each slice by calculating its size or error rate (d'Eon et al., 2021; Singla et al., 2021), but fail to capture any notion of the slices' coherence. The second category of evaluations compare the output slices to known groups where the model underperforms (Sohoni et al., 2020; Eyuboglu et al., 2022; Plumb et al., 2023; Wang et al., 2023). One major limitation of this approach is that we often lack access to the full set of groups where a model underperforms. As such, these works evaluate methods' ability to discover only the same subset of known errors on well-studied benchmarks (Sohoni et al., 2020). The final category of evaluations uses another ML model (in place of a human) to generate a natural language description of each slice (Eyuboglu et al., 2022; Gao et al., 2022). However, recent work has shown that these ML models often generate nonsensical descriptions (Gao et al., 2022). Our study avoids limitations of past automated evaluations by instead running a human evaluation that asks people (rather than a separate ML model) to describe each slice. Our hypothesis validation approach also does *not* require access to the full set of true model errors.

Human Evaluations of Slice Discovery Tools Several works that introduce new slice discovery tools include qualitative evaluations performed by humans. Many such works do not conduct a controlled user study. Instead, the authors themselves describe the semantic features shared by the top 3, 5, or 10 images in each slice (d'Eon et al., 2021; Singla et al., 2021; Eyuboglu et al., 2022; Wang et al., 2023). To our knowledge, the majority of these evaluations do not validate that their descriptions accurately depict groups where the model does underperform. For example, d'Eon et al. (2021) describe the top-10 images in a slice output by their method as "*consisting mostly of images of young children*", and therefore state "*[their algorithm] identified age groups the model performs poorly on*". However, to our knowledge the authors do not validate this claim. Some works use the authors' ability to find a description that matches the majority of images in the slice as a proxy of the algorithms' utility. For example, d'Eon et al. (2021) argue that their method is superior to a competitor because the latter's output slices appear to "*share little in common*".

One of the most similar studies to ours is Singla et al. (2021), where industry data scientists are asked to use slices output by an algorithm to develop hypotheses about where an image classifier fails. Like past work, their study does not validate the correctness of participants' hypotheses. Further, they only evaluate a single slice discovery algorithm.

In contrast to past work, our work does *not* assume that users' hypotheses are de facto correct depictions of true errors. Instead, we validate each user-generated hypothesis by calculating the model's accuracy on a new sample of data. Furthermore, we explicitly study whether output slices' coherence is a valid proxy for tools' utility.

Behavioral Understanding of ML A growing number of works propose tools to facilitate stakeholders' understanding of model behavior. Our work contributes to the growing body of empirical studies that ask users to form hypotheses about where a model underperforms (Singla et al., 2021; Cabrera et al., 2023b; Suresh et al., 2023; Moore et al., 2023). However, relatively few existing studies validate the correctness of users' hypotheses (Wu et al., 2019; Cabrera et al., 2021; Gao et al., 2022). To our knowledge, ours is the first study to evaluate whether slice discovery tools help users form *correct* hypotheses of model behavior.

B. Slice Discovery Algorithms: Extended Descriptions

DOMINO (Eyuboglu et al., 2022) works by first embedding each image using a multi-modal embedding. We use CLIP (Radford et al., 2021) for its demonstrated performance on natural images. Next, DOMINO fits an error-aware Gaussian Mixture Model to model the input embeddings, model predictions, and true labels for each datapoint. DOMINO outputs the top-k mixture components with the largest discrepancy between the model's predictions and true labels. We ran DOMINO using the default hyper-parameters used in their experiments ($\gamma = 10$).

PLANESPOT (Plumb et al., 2023) differs from Eyuboglu et al. (2022) in two ways: First, PLANESPOT uses the final hidden layer activations of the neural network model that we aim to form hypotheses about (instead of a separate pretrained embedding). Second, while PLANESPOT also fits an error-aware Mixture Model, it does so by appending the model's predicted confidence to the model's embedding. PLANESPOT outputs the top-k mixture components that have the largest product of their error rate and number of errors to prioritize large and high-error slices. We ran PLANESPOT using the default hyper-parameters used in their experiments (i.e. running `scvis` using all default hyper-parameters from their python package, and $w = 0.025$).

Our BASELINE algorithm randomly samples from the subset of misclassified images, without replacement. To return k slices that each have m images, we randomly sample m images without replacement from the set of misclassified images that have yet to be added to a slice.

Within each slice output by a slice discovery method, the datapoints are ordered by their component-conditional likelihood, where the "most likely" datapoints are thought to be the "most representative" of the semantic features that unify the slice (Eyuboglu et al., 2022). The datapoints within each slice output by the BASELINE algorithm are ordered arbitrarily.

C. Image Retrieval & Approximation Strategy: Extended

Motivation. As stated in Section 2, one major challenge is that in many settings, we do not have access to Φ, the complete set of images in D that match each hypothesis. Unfortunately, obtaining Φ is often difficult and expensive. For example, recall the scenario where a practitioner hypothesizes that her model underperforms at detecting stop signs for "photos taken in snowy environments". In this setting, the practitioner a-priori does not know which images in the class D match her hypothesis. She could manually review all photos of stop signs and annotate whether they were taken in snowy environments, but this strategy is slow and inefficient.

In our study, we aim to validate the correctness of 180 user-generated hypotheses. Unfortunately, manually reviewing each image within each class D to obtain the full set of images that match each hypothesis is difficult at scale. Thus, we decided to approximate the model's performance gap for each hypothesis using a sample $\hat{\Phi}$ of images that match each hypothesis, where $\hat{\Phi} \subset \Phi$.

Retrieval Method. Our goal is to retrieve a sample $\hat{\Phi}$ of images that match a text description hypothesis t, given a dataset of candidate images D. At a high level, our approximation strategy has three steps:

1. Use CLIP to compute a similarity score between the text description t, and every image in D.

   As proposed by the original paper (Radford et al., 2021), we interpret the *cosine similarity* between each (normalized) text description and image embedding as a measure of the semantic similarity between them.

2. Manually inspect a sample of the top 80 most similar images, and add up to the first 40 images that match the description as belonging to $\hat{\Phi}$. (This step is detailed in Appendix D.)

3. Compare the model's accuracy on $\hat{\Phi}$, to the model's accuracy on another sample $\hat{D}$.

Ablations We tried three versions of the above approximation strategy. The first version (and its results) was presented in the main text. We detail two alternative versions to highlight several subtleties that we discovered when considering how to best validate users' hypotheses.

In summary, ideally we would retrieve a "representative sample" of images $\hat{\Phi}$ that is drawn randomly from Φ, so that the model's performance on $\hat{\Phi}$ is an accurate representation of the model's performance on the entire group Φ. However, we noticed that small changes to our CLIP retrieval strategy changed the types of images that were retrieved. Despite this variance, we found that all of our research hypotheses held across all retrieval strategies. We highlight some of the subtleties we discovered with CLIP retrieval here to share with the research community, as CLIP image retrieval is becoming increasingly common in model debugging work.

Version #1: Compare two samples retrieved by CLIP In Version #1, we use the above strategy with no modifications. For the final Step #3, we compare the model's accuracy on $\hat{\Phi}$ to the model's accuracy on the top-100 images that are most similar to the prompt, "*a photo of [class]*", e.g. "*a photo of broccoli*".

We chose to compare the model's performance to a sample $\hat{D}$ (rather than all of the images belonging to the class) to control for the implicit bias of CLIP's retrieval algorithm. Table 1 shows the model's average performance on the 100 most similar images to the class prompt, vs. all of the images in each class. With the exception of `skis`, for the majority of classes, the model performs significantly better on the 100 most similar images to the class prompt, especially for the `broccoli` class. We hypothesized that this occurs because CLIP is more likely to rate images where the class is is an iconic view (i.e. is not occluded) as being more similar to the class prompt. Thus, because this implicit bias is likely reflected in the sample $\hat{\Phi}$ (i.e. the true class may be more likely for an ML model to detect in the sample retrieved using CLIP), we decided to "control" for this implicit bias by comparing the model's performance to another sample retrieved using a similar prompt with the class name.

Version #2: Compare the sample to the entire class In contrast to Version #1, we also ran an ablation where we approximated the model's performance as the difference in accuracy on $\hat{\Phi}$, vs. the *entire class* D (i.e. the class accuracies under the "Overall" column of Table 1). As expected, because the model had *lower* performance for four out of the five classes, this ablation resulted in a smaller proportion of hypotheses being deemed "correct" for a fixed threshold τ.

Class	**CLIP Sample**	Overall	Δ
airplane	**0.88**	0.76	+0.12
train	**0.81**	0.73	+ 0.08
giraffe	**0.98**	0.83	+ 0.15
skis	**0.62**	0.74	- 0.12
broccoli	**0.88**	0.58	+ 0.30

Table 1. **Class Overall vs. CLIP Sample Accuracy.** The model's recall on the top-100 most similar images to prompt, "*a photo of [class]*" using CLIP (Radford et al., 2021)'s cosine similarity, vs. all of the test set images that belong to each class.

Version #3: Compute contrastive similarity scores Our final ablation strategy modified Step #1, the way we define which images are "most similar" to the hypothesis prompt. Specifically, rather than using the raw cosine similarity score between the individual hypothesis prompt and each image, we instead calculated the relative *likelihood* that each image matched two candidate descriptions: (1) the hypothesis prompt, and (2) the class prompt. We defined the "most similar" hypotheses as those that had the highest predicted probabilities of matching the hypothesis prompt.

For example, if the users' hypothesis was "*a photo of a giraffe and zebra*", then we would calculate the relative likelihood that each image "matched" the hypothesis prompt, vs. the class prompt ("*a photo of a giraffe*").

To calculate the final performance gap, we compare the model's performance on the matching $\hat{\Phi}$ to the entire class D (like Version #2).

We were motivated to try this variant for several reasons. Intuitively, because we were already searching for photos that matched the hypothesis within a dataset of images that all belonged to the same class, including the class's name in the prompt wasn't really helping us narrow down on images that matched the hypothesis. Because past work has shown that CLIP effectively functions as a bag-of-words model (Yuksekgonul et al., 2023), searching with the hypothesis prompt alone may just return images of "*giraffes and zebras*" that look extra giraffe-y (and don't even have the zebras that we want!).

To examine the difference between the default vs. contrastive CLIP retrieval strategy, we ran a small-scale experiment where we retrieved the 80 most similar images for 5 hypotheses, 1 per each class:

- `a photo of water skis`
- `a photo of an airplane with people`
- `a photo of broccoli with other food`
- `a photo of a giraffe with zebras`
- `a photo of many people standing outside a train`

For each hypothesis, we examined a sample of the top-80 most similar images, and labeled all of the images that did or did not match.

We found that the model had a significant difference in accuracy for the two retrieval strategies: on average, the model had 35% accuracy on the samples retrieved contrastively, vs. 65% accuracy using the default strategy. When we looked only at the examples that we labeled as *matching* the hypothesis, the gap was more narrow but still existed: the model had 30% accuracy on the contrastively retrieved images, vs. 47% accuracy on the default strategy.

However, the contrastive retrieval strategy offered one major benefit: a much greater percentage of the images that were retrieved contrastively matched the hypothesis text description (75% vs. 48% for the default strategy). In conclusion, retrieving contrastively to the class prompt was more likely to return matching images; but may result in an under-estimation of the model's true performance on all in-distribution images that match.

In conclusion, there are several reasonable strategies that have various pros and cons to retrieve a sample of images that matches a text description. Specifically, while retrieving images contrastively to the class prompt does result in a higher-quality sample, the model has significantly lower accuracy on the retrieved sample. Future work should continue to critically examine why and how the retrieval process used to find new images that match a hypothesis may mischaracterize the model's true behavior on the group.

D. Labeling Images that Match User Hypotheses

Figure 5. Example images that a user selected as matching their hypothesis, "*a photo of people feeding a giraffe*"

To retrieve the sample $\hat{\Phi}$, we had to come up with a reproducible and consistent process to determine whether a new image matched the users' text description. To do this, we created a *labeling guide* inspired by (Shankar et al., 2020; Vasudevan et al., 2022) describing the criteria we used to determine whether an image matched each hypothesis. We will publicly release our labeling guide at `https://github.com/TODO`.

To ensure that our own labeling process was consistent with the users' intent, we checked that the criteria in our labeling guide were consistent with the user's labeling process by referencing the (up to 20) images that the user selected as matching their description. We only added labeling criteria that were consistent with the users' own labels. We (the annotators) were blind to each group description's associated condition (algorithm) and the model's error rate on the selected images to avoid biasing our labeling.

We walk through an example to illustrate our labeling process. Consider the hypothesis, "*a photo of people feeding a giraffe*". The user selected the images shown in Figure 5 as matching their hypothesis. Given a sample of new images that all had giraffes, we had to determine which images belonged vs. did not belong to the users' group. For this hypothesis, we came up with the following guidelines:

- At least one person must be present in the image (i.e. images of giraffes eating with no people in them do not belong to the group).
- Either (1) the person must be holding some food item (as in the bottom left image), or (2) the giraffe must be holding some food item in its mouth (as in the top left image) to belong to the group. Images where people are standing near a giraffe, but there is no food, do not count.

Note that the above guidelines are consistent with the users' own labels of the images that matched their hypothesis.

E. Study Design: Extended

In this section, we provide an extended description of our user study design.

E.1. Domain & Model

To study whether slices can help humans understand model behavior, we must first select a domain, prediction task, and model. We used data from MS-COCO ("COCO") (Lin et al., 2014), a large object detection dataset, for its accessibility to a wide audience. COCO contains photos with 91 different object types "*that would be easily recognizable by a four-year-old*" (Lin et al., 2014) in everyday natural scenes. COCO is a multi-label classification task where a single image can have several objects present. We defined a custom 15%-10%-75% train-validation-test split so that we could use a larger held-out test set as input to the slice discovery algorithms. We fine-tuned a pretrained ResNet-18 model using the training set (details in Appendix G), performed model selection using the validation set, and ran the slice discovery algorithms on the held-out test set. Because we aim to evaluate how well slice discovery tools can detect naturally occurring errors, we did not modify the dataset or model training process to synthetically induce specific errors.

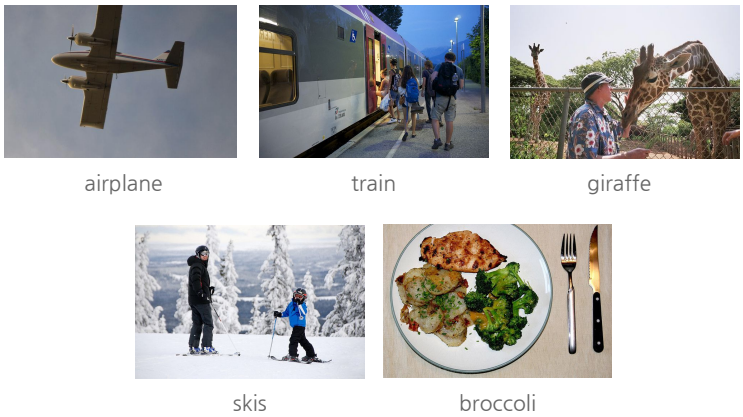

Figure 6. Example photos from the 5 selected COCO object classes (Lin et al., 2014).

E.2. Computing Slices

We generated 60 total slices (20 slices per each of the 3 algorithms) to show users. Rather than run each slice discovery algorithm on the entire COCO test set (which contains 91 object types), we followed past evaluations (Gao et al., 2022; Plumb et al., 2023) and ran each algorithm on subsets of the data that all have the same object. Thus, each hypothesis describes a group where the model has low *recall* (i.e. failed to detect the object). We used each slice discovery algorithm to return the top $k = 4$ slices for 5 different objects (Figure 2, Top), chosen randomly from a list of candidate objects where the model had at least 50% recall: `airplane`, `train`, `giraffe`, `skis`, and `broccoli`.

E.3. Participants & Recruitment

We recruited 15 subjects that had self-reported "intermediate knowledge in machine learning (ML) or computer vision (CV)" (i.e. had taken a graduate-level course or had practical work experience in ML, AI, or CV) using university mailing lists. All participants were students enrolled in a full-time degree program in computer science. Participants were compensated with a $20 gift card, and reported spending 30 (min) to 55 (max) minutes participating.

E.4. Study Procedure

The study was approved by an Institutional Review Board (IRB) process and was conducted asynchronously online. Participation was voluntary and users were shown a consent form before participating. We began each study by presenting the user with a description of the study task and walk-through of the study interface. After completing the walk-through, the user was shown information about 12 different slices belonging to 3 different object classes. The user completed

a questionnaire that asked them to formulate a behavioral hypothesis for each slice. We detail each phase of the study procedure below.

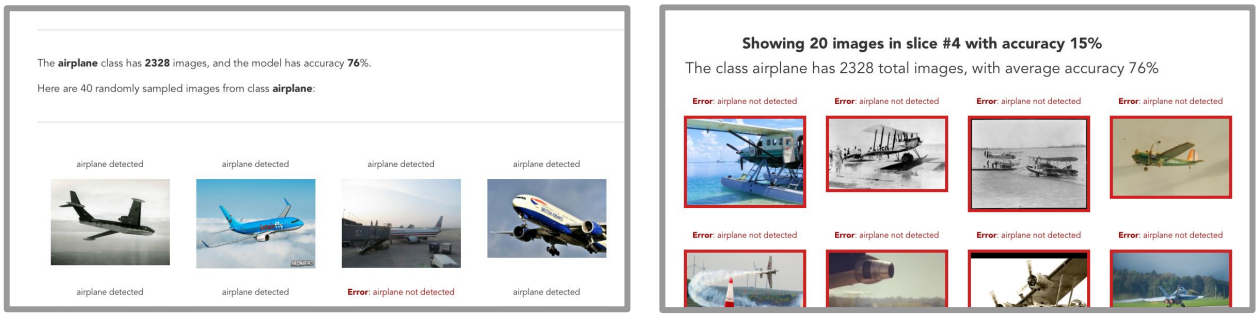

Figure 7. UI screenshots of the class overview (Left) and slice overview (Right). Errors (where the model failed to detect the object) have red borders. (Left, Top) The class overview shows the total number of images in the test set that belong to the class and the model's average accuracy on these images. (Left, Bottom) A random sample of 40 images from the test set. (Right, Top) The slice overview shows model's average accuracy on the top-20 images belonging to the slice and the entire test set. (Right, Bottom) The top-20 ordered images that belong to the slice.

Instructions In the study walk-through, we introduced the user to the object detection task by showing them example images and model predictions from a randomly selected class ("tennis racket"). We defined and motivated the slice discovery task by listing several reasons why a practitioner may wish to discover subgroups where a model underperforms. We showed the user two example slices for the "tennis racket" class and example behavioral hypotheses (i.e. descriptions of the semantic features that unite each slice) that could correspond with each slice. We provide screenshots of the task instructions in Appendix H.

Class Overview Each user was asked to form hypotheses for slices corresponding to 3 different object classes. For each class, the user was first shown a *class overview* (Figure 7, Left) with information about and example images belonging to the class. We presented the class overview before presenting the slices to give the user basic context about the COCO dataset (i.e. examples to illustrate the variety of images that belong to each class) that stakeholders in practice would already have for their domain.

Slice Overview & Questionnaire For each slice, the user was shown a *slice overview* (Figure 7, Right) that displays the model's performance on the top 20 images that belong to the slice. The user was asked to use this information to complete the *slice questionnaire*, which asked them to (1) write down a behavioral hypothesis of the underperforming subgroup corresponding to the slice, (2) select all images in the slice that belong to this subgroup, and (3) rate how difficult it was to describe the slice. We provide the complete questionnaire in Figures 11 and 12.

Visualizing each slice. We decided to control for how we visualized the model's performance on each slice by presenting the top-20 images only for all conditions. We designed our slice overview to be as similar as possible to how past work (d'Eon et al., 2021; Eyuboglu et al., 2022) presents individual slices. While these works are not prescriptive that this is how these slices *should* be visualized, we believe there is value in validating the extent to which these algorithms achieve their proposed goals using the authors' visualization. We limit the number of images shown to 20 because showing more images may cause information overload and increase the time required to complete each questionnaire.

Eliciting a hypothesis for each slice. Past works assume that *all images* in each computed slice correspond directly to a *single* subgroup where the model underperforms (d'Eon et al., 2021; Eyuboglu et al., 2022; Plumb et al., 2023). We explicitly evaluate this assumption and ask users to form a single behavioral hypothesis for each slice. Specifically, we instruct users to *describe* the slice to the best of their ability by writing a group description that matches as many images in the slice as possible. Users are told that some slices may be noisy or incoherent, and that in some cases it may be difficult to find a single description that matches all of the images in the slice. We provide users with further guidance and example hypotheses detailed in Appendix H.

F. Pilot Studies

We ran several initial pilot studies to elicit feedback on our study instructions, interface, and design. Our pilot studies demonstrated the importance of the instructions used when eliciting users' hypotheses. Specifically, in an early pilot study, we did not prompt users to write down a single subgroup corresponding to each slice, but instead allowed them to freely describe where they believed the model underperformed. We found that when given this freedom, users often wrote down hypotheses that were the union of two distinct groups (e.g. "*silhouetted giraffe OR giraffe with striped animal*". This behavior, while interesting, contradicted past works' assumption that each individual slice should only capture a single group where the model underperforms. In the end, we decided to change our study instructions to elicit only a single group for each slice.

Our initial pilot study results informed our research hypotheses. Specifically, we were surprised to observe that the majority of users wrote down different hypotheses when we showed them the same slice in an early pilot study, where we showed the users only a single slice per each class. This observation inspired our final research hypotheses.

G. Training Details

We fine-tune a pretrained (on ImageNet) ResNet-18 from the `torchvision.models` package. We use the fine-tuning procedure proposed by Kumar et al. (2022), where we first linear probe (i.e. hold all but the last linear layer as fixed) before we fine-tune (i.e. optimize over all of the network weights). We train using Adam with learning rate 0.001. We perform model selection by choosing the model with the best validation set cross-entropy loss.

H. Task Instructions

H.1. Task Description

Below, we paste excerpts from the task description that was provided to users in the tutorial.

Machine learning models can have blindspots, which occur when the model has lower accuracy on a coherent group of images (i.e., the images in the group are united by a human-understandable concept).

Example. Given a dataset of images with tennis rackets, the model has much lower accuracy on images of "tennis rackets without people" (accuracy 41.2%), compared to images of "tennis rackets with people" (accuracy 86.6%). Therefore, the model has blindspot "tennis rackets without people", because it has lower accuracy on images belonging to the group compared to images outside of the group.

In practice, after a model developer has trained a new object detection model, while the model may have blindspots, the model developer does not know what the model's blindspots are.

Why discover blindspots?: There are several reasons why one may wish to discover an ML model's blindspots. For example,

- Blindspots where the model underperforms on specific groups can contribute to algorithmic bias and cause downstream harm.
- Knowledge of where the model underperforms can inform decisions about deployment and actions model developers can take to fix (e.g. re-train) the model.

Blindspot discovery methods are algorithms designed to help humans discover model blindspots. Given a dataset of images, the goal of a blindspot discovery method is to output "slices": groups of images that are both (1) high-error, and (2) coherent. Each slice is designed to correspond to a different true model blindspot.

However, the slices returned by these algorithms may not always be coherent. They may be noisy (e.g. contain images that differ from the rest of the group), or fail to capture a coherent concept altogether.

In this study, you will be shown information about the model's performance on several slices output by a blindspot discovery algorithm.

For each slice, your primary goal is to describe the slice to the best of your ability by writing down a text description that:

1. captures one distinct concept,
2. that matches as many images in the slice as possible

What does it mean to "describe" the slice?:

- Your description should capture only one concept, even if it appears as though there are several different concepts represented in the slice.
  - For example, if 5 of the images in the slice are of "tennis rackets with people" and 15 of the images in the slice are of "tennis rackets with dogs", you should write down "tennis rackets with dogs" (as this description matches more of the images in the slice).
- The slice might be noisy or incoherent: it may be difficult to find a single slice description that matches all of the images in the slice. In these scenarios, you should try to write down a description that describes as many of the images as possible - preferably, at least 5 images (but the more the better!).
- Your slice description should be as clear, not subjective, and un-ambiguous as possible. Another person should be able to read what you write down, and quickly determine if a new image belongs to the group.

Some example slice descriptions for the tennis racket example are:

- tennis rackets without people
- tennis rackets on clay courts
- tennis rackets without a tennis ball
- photos of tennis rackets taken inside of a residence

Examples of bad slice descriptions:

- tennis rackets with dogs or tennis rackets with people. Problem: This description is the "or" of two distinct concepts, when you are only supposed to have one! Potential fix: "tennis rackets with dogs"
- small tennis rackets. Problem: This description is ambiguous: what is a "small" vs. "big" tennis racket? Potential fix: "tennis rackets designed for children"

Why describe each slice?

```
Because each slice was designed to correspond to a true model blindspot, being
able to describe (in words) the true blindspot enables stakeholders to understand
and communicate about actions they can take to address it.

Because blindspot discovery algorithms only return groups of points, they were
designed with the goal that humans can look at their output and make sense of
what coherent concepts are captured by each slice.  For example, discovering
that "the model has accuracy 35% on all images in the dataset of tennis rackets
without people" is much more informative than, "the model has accuracy 30% on
this set of 20 images returned by a blindspot discovery algorithm".
```

H.2. Task Instructions

See Figure 10.

I. Identifying Distinct Hypotheses: Extended

For each slice, we asked two annotators to label whether pairs of hypotheses were *synonymous*, i.e. describe identical groups of points. We presented the annotators with the following instructions:

```
In this study, you will be shown lists of different group descriptions.  Each
description was written by a human, and describes a group of images.

Your goal is to identify which descriptions are synonyms.  Synonymous
descriptions may differ syntactically, but describe the same group of images.  If
an image belongs to 1 group, it also will belong to all of its synonymous groups.

For example, the following two descriptions are synonyms, even though they differ
syntactically:

• D1:  "black-and-white photos of giraffes"
• D2:  "giraffes in greyscale"

The following two descriptions, while similar, are not synonyms, as there may be
some images that would belong to D2 but not D1.

• D1:  "a photo of children eating broccoli"
• D2:  "photos of broccoli and small children"
```

Together, two annotators together annotated 48 (out of 120 possible) unique pairs of hypotheses as synonyms. They disagreed (i.e. only one annotator marked the pair as being a synonym) on 22 of the pairs. We define a pair of hypotheses as *distinct* if *neither annotator* noted that the pair was synonymous. In other words, if at least one annotator stated that the pair was synonyms, then we do not consider the hypotheses as being distinct.

J. Results: Extended

Below, we detail the results of the statistical tests discussed in the main text, and present results from additional experiments when relevant.

J.1. Correctness (Extended)

Below we present the complete results of the statistical test for **H1**.

We ran an ANOVA test with Tukey post-hoc tests for multiple comparisons. We compared the indicator for whether each hypothesis was correct with gap threshold $\tau = 0.2$ for each algorithm condition (i.e. BASELINE, Domino, PLANESPOT)

corresponding to the hypothesis. We excluded hypotheses where we failed to find at least 15 matching images (to approximate the performance gap) from our analysis. We retained 51, 41, and 43 rows corresponding to the BASELINE, PLANESPOT, and DOMINO conditions respectively. Figure 3 visualizes the mean and standard error of the correctness indicators.

The value of the ANOVA F statistic was 6.896, and we found a statistically significant difference between the conditions ($p = 0.0015$). We report each pair-wise p-value in Table 3.

Conditions	p-value
DOMINO - BASELINE	**0.0009***
PLANESPOT - BASELINE	0.3775
PLANESPOT - DOMINO	0.0714

Table 2. **Hypothesis Correctness.** Results of Tukey post-hoc tests for pair-wise comparisons. Significant p-values are denoted with an asterisk (*).

J.1.1. ABLATION: PERFORMANCE GAP THRESHOLD

We repeat the same hypothesis testing procedure where we ablate the performance gap threshold τ used to calculate the "correctness" indicators. Figure 8 displays how the average proportion of correct hypotheses changes as we increase the performance gap threshold τ. For all $\tau \leq 0.5$, we observe that a greater proportion of user hypotheses are "correct" for slices output by the DOMINO, then PLANESPOT, then BASELINE algorithms. Table 8 shows the p-values of pairwise comparisons between conditions at different thresholds τ. We observe a statistically significant difference between the DOMINO and BASELINE conditions for all thresholds $\tau \leq 0.4$. There is a statistically significant difference between the PLANESPOT and BASELINE conditions for $\tau = 0.3$ only. There is no significant difference between the PLANESPOT and DOMINO conditions for any threshold.

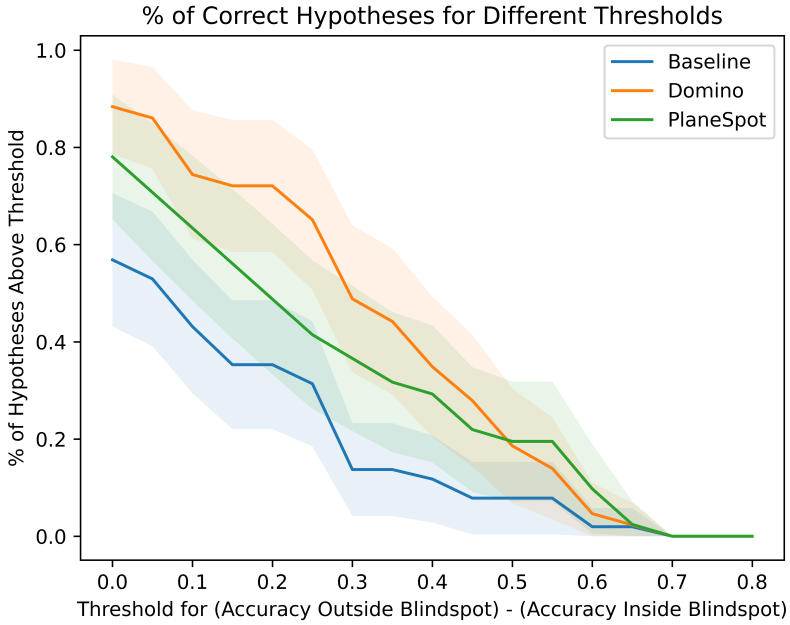

Figure 8. **Hypothesis Correctness.** The proportion of "correct" hypotheses per condition (y-axis), when we vary the performance gap threshold τ (x-axis). The shaded region displays the 95% confidence interval (defined as $\pm 1.96\times$ the standard error) for each group.

Conditions	$\tau = 0.1$	$\tau = 0.15$	$\tau = 0.2$	$\tau = 0.25$	$\tau = 0.3$	$\tau = 0.35$	$\tau = 0.4$
DOMINO - BASELINE	**0.0055***	**0.0009***	**0.0009***	**0.0027***	**0.0006***	**0.0030***	**0.0247***
PLANESPOT - BASELINE	0.1112	0.1007	0.3775	0.5797	**0.0418***	0.1290	0.1222
PLANESPOT - DOMINO	0.5444	0.2816	0.0715	0.0671	0.4217	0.3976	0.8154

Table 3. Results (p-values) of Tukey post-hoc tests for pair-wise comparisons. Significant p-values are denoted with an asterisk (*).

J.2. Number of matching images (extended)

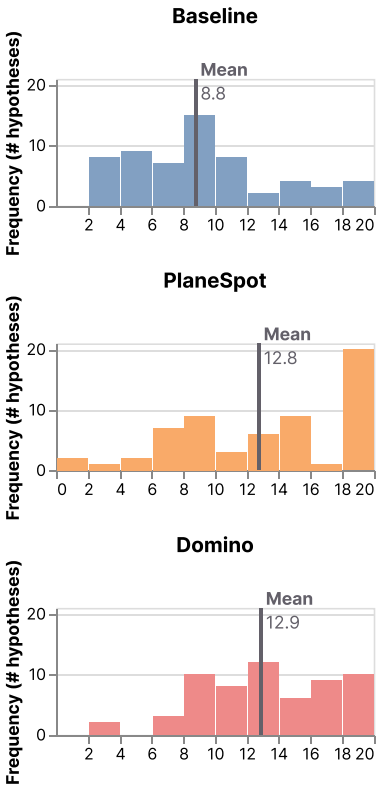

Figure 9. **Number of Matching Images.** Histograms of the number of images in each slice that match the user's hypothesis (out of the top 20), stratified by each condition (slicing algorithm). The mean number of matching images for each condition is denoted by a vertical line. We find that on average, users selected more images as "matching" their hypothesis for slices output by PLANESPOT and DOMINO.

Figure 9 shows the mean, standard error, and distribution of the number of images that match each hypothesis. The ANOVA F statistic was 13.895, and $p < 0.0001$. We report each pair-wise p-value in Table 4.

Conditions	p-value
DOMINO - BASELINE	**0.0000***
PLANESPOT - BASELINE	**0.0000***
PLANESPOT - DOMINO	0.9876

Table 4. **Number of Matching Images.** Results of Tukey post-hoc tests for pair-wise comparisons. Significant p-values are denoted with an asterisk (*).

J.3. Self-reported difficulty (extended)

Figure 4 shows the average self-reported difficulty of describing each slice, and a histogram of users' responses for each slice. We report the W-statistic and p-value for each pair-wise comparison in Table 5.

Conditions	W-statistic	p-value
DOMINO - BASELINE	$W = 2798.5$	$\mathbf{0.0000}^*$
PLANESPOT - BASELINE	$W = 2661.5$	$\mathbf{0.0000}^*$
PLANESPOT - DOMINO	$W = 1695$	0.5591

Table 5. **Self-Reported Difficulty.** Results of the pair-wise Mann-Whitey tests. Significant p-values (accounting for the Bonferroni correction) are denoted with an asterisk (*).

J.4. Correctness: CLIP Retrieval Ablation

We also calculated the proportion of correct hypotheses for the two alternative approximation strategies detailed in Appendix C and ran our statistical tests using a correctness threshold of $\tau = 0.2$. When we retrieved images *contrastively*, we retained 160 out of 180 total hypotheses (89%). We found that for all retrieval strategies, a consistent trend holds: a larger proportion of DOMINO hypotheses are correct compared to PLANESPOT, which outperforms BASELINE (Table 7). The difference between the DOMINO and BASELINE conditions is significant for all thresholds. The DOMINO condition significantly outperforms the PLANESPOT condition for the contrastive retrieval strategy only.

The percentage of all hypotheses that are correct for a fixed threshold τ varies across approximation strategies. Approach #2 appears to be more conservative in that only 37% of all hypotheses are correct, which is significantly fewer than 68% of all hypotheses for Approach #3. This difference aligns with our exploratory analyses in Appendix C, in which we observed that retrieving images contrastively tends to result in a less canonical (and thus more difficult) sample for a model to do well on. In summary, while our finding of the relative ranking across conditions holds consistently for all approximation strategies, our results point to the difficulty of obtaining a representative sample $\hat{\Phi}$ using existing image retrieval tools.

Finally, our finding that slice coherence is uncorrelated with hypothesis correctness (**H4**) holds for all three approximation algorithms (Table 6).

Approximation Strategy	r_s	p
Approach #1	0.07	0.5027
Approach #2	0.08	0.4728
Approach #3	0.07	0.5020

Table 6. **Coherence vs. correctness for different performance gap approximation strategies**. Reports the Spearman's rank correlation coefficient r_s and p-value for the number of matching images (IV) vs. the indicator for whether the hypothesis is correct (DV), defining "correctness" using the three different approximation strategies detailed in Appendix C.

Condition	Approach 1	Approach 2	Approach 3
All	0.51 (0.043)	0.37 (0.04)	0.68 (0.04)
BASELINE	0.35 (0.067)	0.20 (0.06)	0.48 (0.07)
PLANESPOT	0.49 (0.079)	0.41 (0.08)	0.67 (0.07)
DOMINO	0.72 (0.069)	0.53 (0.08)	0.89 (0.04)

Table 7. **Correctness with CLIP Retrieval Ablations.** The percentage of hypotheses per condition that are "correct" (with standard errors) when we use samples $\hat{\Phi}$ retrieved by three different strategies detailed in Appendix C. We use a performance gap threshold $\tau = 20\%$. Percentages are calculated for the subset of hypotheses that we have a sufficiently large number of examples (i.e. at least 15 matching images) to approximate the performance gap.

Conditions	Approach 1	Approach 2	Approach 3
DOMINO - BASELINE	$\mathbf{0.0009}^*$	$\mathbf{0.00018}^*$	$< \mathbf{0.0001}^*$
PLANESPOT - BASELINE	0.3775	0.0690	0.0817
PLANESPOT - DOMINO	0.0714	0.4661	$\mathbf{0.0321}^*$

Table 8. **Correctness with CLIP Retrieval Ablations.** Results of Tukey post-hoc tests for pair-wise comparisons, comparing the proportion of correct hypotheses with threshold $\tau = 0.2$. Significant p-values are denoted with an asterisk (*).

K. Discussion: Extended

K.1. Interpreting the experimental results

We return to our study goals to place our results in conversation with past work on slice discovery.

Benchmarking existing tools Overall, we found significant differences across measures between the DOMINO and PLANESPOT conditions, relative to the naive baseline algorithm. Specifically, while DOMINO and PLANESPOT slices had a significantly higher average number of matching images and were significantly easier to describe relative to the baseline condition, supporting **H2** and **H3**, only DOMINO had a statistically significant difference in the proportion of correct hypotheses. While PLANESPOT had a higher proportion of correct hypotheses than the BASELINE condition, the difference was not statistically significant for the majority of performance gap thresholds τ (Appendix J.1.1).

In summary, our results provide preliminary evidence that existing tools may offer some benefit relative to a naive workflow of inspecting and trying to make sense of a random sample of errors. However, the slices output by existing tools do not *always* help users form correct hypotheses. For example, only 49% of users' hypotheses corresponding to slices output by PLANESPOT described groups where the model performed at least 20% worse. Thus, existing tools do have the potential to mislead users to develop false beliefs if used as proposed by past work.

Coherence does not imply correctness Our study calls into question previous assumptions about slice *coherence* and hypothesis *correctness* (**H4**). A number of works have implied that if the user can describe all the images in a slice, that the model underperforms on all images that match their description. Our finding that there is no significant association between slice coherence and hypothesis correctness shows that this assumption is misleading.

While it may seem counter-intuitive, we are *not* surprised that existing slice discovery tools tend to underrepresent the model's performance on coherent subgroups. We see this behavior as a feature (rather than a bug) resulting from the way that existing tools are designed to optimize to return high-error subsets. Existing tools are incentivised to output misleading slices that contain a misrepresentative sample of high-error images *within* each group, even in cases where the model doesn't actually perform worse on the *entire* group. Evaluations that only consider the coherence of the output slices fail to account for whether the slice is a representative sample of the model's performance on the entire group.

This finding has significant implications for researchers, who should center hypothesis correctness (rather than slice coherence) when evaluating their tools. But perhaps more importantly, this finding is significant for *users* of these tools. While present-day tools can help users form hypotheses, users should take caution to know that their hypotheses, no matter how aligned with the slices they may be, may not necessarily be correct. Thus, validating behavioral hypotheses is not only important for researchers, but also for users.

K.2. Design Opportunities

Our findings point to several design opportunities for ML researchers and UI designers. We highlight a few exciting directions for future work below.

Supporting hypothesis formation In the hypothesis formation step, stakeholders make sense of a large amount of information (e.g. the images belonging to each slice) to form hypotheses about model behavior. However, the human factors that affect participants' hypothesis formation process have been understudied.

One example dimension that has been neglected by past work is how the output slices should be *visualized* and presented to users. When asked if there was "any other information that they were not shown that [they] believe would have helped [them] complete the quesionnaire", several users expressed a desire to see the model's performance on a larger set of examples

beyond the top-20 images belonging to the slice. One user wrote, "*it would have been helpful to retrieve other images from the dataset during slice labeling* (hypothesis formation)" to "*see how good my label* (hypothesis) *was*". We envision several alternative ways to present each slice beyond the static grid used in our slice overview. For example, one could design novel interactive workflows that allow users to explore the model's performance on multiple slices simultaneously (Bertucci et al., 2022; Cabrera et al., 2023a), or compare the images within to images outside of each slice. Designing an appropriate visualization that accounts for the cognitive load and biases of the user (Gajos & Chauncey, 2017; Cabrera et al., 2023b) is an important direction for future work.

Prioritizing hypothesis correctness Our study shows that slices that appear to be coherent can mislead users to develop incorrect beliefs about where their model underperforms. Thus, we urge researchers to prioritize developing tools that help users form *correct* hypotheses. One problem we identified is that existing slice discovery tools are incentivised to output high-error samples that may not capture the full diversity of images that belong to each group. Future work could study how to output slices that contain a more representative sample of each group.

Towards real-time hypothesis validation Another promising direction is to develop tools that allow users to validate their own hypotheses in real time. Such tools would enable users to iteratively refine their hypotheses (i.e. explore multiple possible errors that could correspond to a slice), and serve as a sanity check before investing in expensive downstream actions based on false beliefs. In their recent literature review, Cabrera et al. (2023b) identified the lack of existing tools to support hypothesis validation as a "significant gap". Most existing workflows to help users collect evidence to validate their hypotheses often prioritize retrieving examples that are most *similar* to those that support the users' hypothesis (Cabrera et al., 2023b; Gao et al., 2022; Suresh et al., 2023). In contrast, we believe a promising direction is to prioritize retrieving *counter-evidence*: examples that contradict the users' hypothesis. We believe that counter-evidence can help combat stakeholders' potential confirmation bias and help them more quickly iterate on their hypotheses.

Interactive workflows for slice discovery We encourage researchers to re-imagine where and how automation can help users discover underperforming groups. Existing slice discovery tools are meant to be run once, and then a stakeholder must make sense of the groups of datapoints they output. One could imagine more bespoke and interactive workflows that better utilize stakeholder's domain knowledge to define coherent subsets of data, or leverage automation to help stakeholders refine their hypotheses. As one example, a stakeholder could guide a clustering algorithm using their contextual understanding of semantic similarity (Rajani et al., 2022; Cabrera et al., 2023a; Suresh et al., 2023) rather than simply looking at the output clusters. More broadly, how can we design algorithmic tools as an aid for the *human task* of hypothesis formation?

K.3. Limitations

The participants in our study were students with intermediate knowledge of machine learning, and we studied a model trained using data from a simple object detection task. Experts such as clinicians (Gaube et al., 2023) or content moderators (Suresh et al., 2023) who have a deeper understanding of their data may interpret the output slices differently. For example, they may be able to use their domain knowledge to better characterize the semantic features shared by the examples in each slice. We encourage future work to study how domain experts form hypotheses using slice discovery tools.

While our work is an important step forward towards prioritizing hypothesis validation, we note that validating natural language hypotheses is difficult, and several open challenges remain. Because retrieving all of the images that match each text description hypothesis is difficult and expensive, we follow Gao et al. (2022) and approximate the model's performance on the group by retrieving a sample of matching images. Unfortunately, the retrieved sample may not be representative of the model's performance on all images in the group; thus, we compare the model's performance to another retrieved sample to account for the implicit bias of our retrieval process (details in Appendix C). For these reasons, our calculated performance gaps are only an *approximation* of the model's performance on each group. Despite these limitations, we found that all of our experimental findings were consistent across a range of ablations to our approximation strategy (Appendix J.4). We believe that identifying inexpensive ways to retrieve a sample that matches a natural language hypothesis is an important direction for future work.

Finally, our study focuses on the task of describing where the model underperforms *in natural language*. We asked users to describe the underperforming groups in words because doing so is a prerequisite for several downstream actions one might take to address the behavior, and more broadly for communicating about the behavior with a wider set of stakeholders. We acknowledge that natural language hypotheses are imperfect for many use cases due to their implicit

subjectivity or under-specification in some contexts. For example, if a practitioner hypothesizes that her object detection model underperforms at detecting stop signs in photos where they are "*far away from the camera*", determining which photos qualify is under-specified from her description alone (just how far away is "far away"?). Developing a more formal, yet simultaneously accessible "domain-specific language" (Wu et al., 2019) for users' hypotheses is an open direction for future work.

Figure 10. A screenshot of the instructions shown to participants before they complete the slice questionnaires for a new object class.

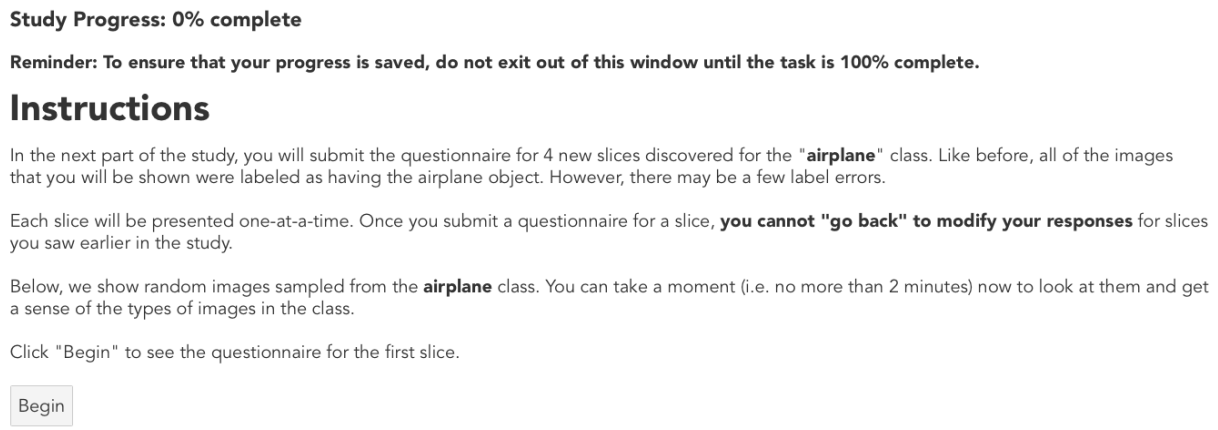

Figure 11. A screenshot of the slice questionnaire.

Questionnaire

1. **Describe** the slice.

Your answer should describe *as many images in the slice as possible*.

2. **Select** (by left-clicking) **all images** in the slice that match your description.

3. Rate how difficult it was to describe the slice on a scale of 1 (Very Easy) to 5 (Very Difficult).

- 1 (Very Easy)
- 2
- 3
- 4
- 5 (Very Difficult)

4. What do you approximate the model's **accuracy** is on types of images that match your slice description from Q1? Enter a percentage between 0 - 100.

100

5. Rate your confidence in your above accuracy estimate on a scale of 1 (not at all confident) to 5 (very confident).

- 1 (Not at all confident)
- 2
- 3
- 4
- 5 (Very confident)

Figure 12. A screenshot of the tooltip text presented to users who click on the (?) icon next to Q1.

Reference: Describing Slices

What does it mean to "describe" the slice?

- Your description should capture only **one** concept, even if it appears as though there are several different concepts represented in the slice.
  - For example, if 5 of the images in the slice are of "tennis rackets with people" and 15 of the images in the slice are of "tennis rackets with dogs", you should write down "tennis rackets with dogs" (as this description matches more of the images in the slice).
- The slice might be noisy or incoherent: it may be difficult to find a single slice description that matches all of the images in the slice. In these scenarios, you should try to write down a description that describes *as many of the images as possible* - preferably, at least 5 images (but the more the better!).
- Your slice description should be as **clear, not subjective, and unambiguous** as possible. Another person should be able to read what you write down, and quickly determine if a new image belongs to the group.

  Some example slice descriptions for the tennis racket example are:
  - tennis rackets without people
  - tennis rackets on clay courts
  - tennis rackets without a tennis ball
  - photos of tennis rackets taken inside of a residence

close