Active Learning Helps Pretrained Models Learn the Intended Task

Anonymous Author(s)
Affiliation
Address
email

Abstract

Models can fail in unpredictable ways during deployment due to *task ambiguity,* when multiple behaviors are consistent with the provided training data. An example is an object classifier trained on red squares and blue circles: when encountering blue squares, the intended behavior is undefined. We investigate whether pretrained models are better active learners, capable of disambiguating between the possible tasks a user may be trying to specify. Intriguingly, we find that better active learning is an emergent property of the pretraining process: pretrained models require **up to 5× fewer labels** when using uncertainty-based active learning, while non-pretrained models see no or even negative benefit. We find these gains come from an ability to select examples with attributes that disambiguate the intended behavior, such as rare product categories or atypical backgrounds. These attributes are far more linearly separable in pretrained model's representation spaces vs non-pretrained models, suggesting a possible mechanism for this behavior.

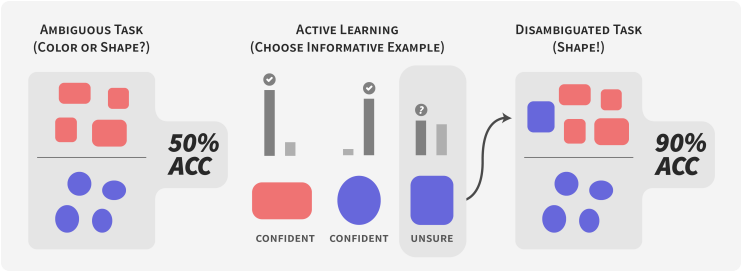


Figure 1: **Active learning can resolve task ambiguity in datasets.** Here, the provided training data leaves the model unsure of the intended task: is it to predict the shape or the color of the object? Pretraining enables models to identify and weigh various rich features, eliciting labels from informative examples (e.g. blue squares) that clarify the user's intention.

1 Introduction

Modern pretrained models can be adapted to new tasks with remarkably little data, enabling downstream applications for tasks with only tens or hundreds of examples [8, 45]. However these low-data applications magnify a fundamental problem in machine learning—namely, that datasets are often incomplete proxies for the desired behavior. For example, a sentiment classifier may behave unpredictably during the holidays if its training data lacks examples of toy reviews. Likewise, an object classifier may struggle to classify objects in atypical environments (e.g. camels in the Arctic) if the training data does not make it adequately clear which features are salient for the task. While

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

these dataset flaws can be solved by labeling better examples, identifying such examples can be challenging, especially as the nature of these flaws may not be known in advance.

One way to describe these kinds of challenges is through the lens of *task ambiguity*: the failure of the training data to fully specify the user's intended behavior for all possible inputs. We consider whether pretrained models can automatically resolve their own task ambiguity through *active learning* (AL). In principle, AL allows models to resolve task ambiguity by identifying examples whose labels would be informative; for example, in Figure 1, the provided training data only contains red squares and blue circles, leaving the intended behavior for blue squares ambiguous. Asking for labels of blue squares resolves this ambiguity. This kind of interaction could clarify the intended behavior for different kinds of examples without the expectation that model developers must anticipate all potential gaps in the model's abilities.

However, AL has often seen limited success in practice. In traditional settings with smaller, non-pretrained models, several problems limit the effectiveness of AL, including label noise, examples that are challenging for models to learn, and a lack of generalizability of AL heuristics [37, 29]. But pretrained models possess several strengths which may enable them to overcome these challenges: they extract high-level semantic features, such as shape and color [43, 22], which can be used to identify informative examples, and they produce more calibrated uncertainty estimates which can be used for selecting ambiguous inputs [24, 14].

We consider the use of active learning (AL) on a range of real-world image and text datasets where task ambiguity arises. We compare several AL acquisition functions against a random-sampling baseline, and compare the difference in performance with and without the use of pretrained models. Our **contributions** demonstrate that:

1. Pretrained models trained with AL can select examples that resolve task ambiguity in the finetuning data
2. The resulting accuracy gains can be quite large in practice: up to $5\times$ reduction in labeled data points for the same performance, or +11% absolute gain for the same labeling budget
3. This ability to actively learn is an emergent property of pretraining: AL has a neutral or even harmful effect on non-pretrained models

2 Method

We study the pool-based active learning (AL) setup common in the literature [51], where we have a (possibly pretrained) model $\mathcal{M}$, a small *seed set* of training data $\mathcal{S} = \{(x_i, y_i)\}$, and a larger *pool* of unlabeled data $\mathcal{P} = \{x_i\}$. The AL procedure proceeds as follows: first, finetune $\mathcal{M}$ on $\mathcal{S}$ until convergence; then, select points x_i from $\mathcal{P}$ that are deemed *most informative* according to an acquisition function $a(x; \mathcal{M})$, obtaining the corresponding labels y_i, until some budget k of data points is exhausted. This newly labeled batch $\mathcal{B} = \{(x_i, y_i)\}$ is then added to the existing data $\mathcal{S}$, and the original model is retrained on $\mathcal{S}$ for the next acquisition step. This process is repeated for a fixed number of acquisition steps.

For our acquisition function, we adopt the classic *uncertainty sampling* approach to AL, in particular the *least confidence* heuristic, where we acquire points for which our model is *least confident* in its predicted label [51]. Specifically, treating the outputs of the model $\mathcal{M}$ as a probability distribution[1] over possible labels $p(y \mid x; \mathcal{M})$, we define the acquisition function to be

$$a(x; \mathcal{M}) = -\max_i p(y_i \mid x; \mathcal{M}) \tag{1}$$

This *least confidence* heuristic has shown to be simple and effective in a variety of settings [51, 23, 39], and we similarly find good results here (we refer to least confidence sampling as "uncertainty sampling" except in Section 5.2 where we explore other uncertainty-based acquisition functions including *entropy* and *margin* sampling).

On top of this standard AL pipeline, we propose the following change to improve the practical applicability of pretrained models in data-scarce settings:

[1] While in general there is no guarantee that this probability distribution will be well-calibrated, recent work has found that pretraining improves model calibration across a variety of settings, including on out-of-domain data [14, 24].

Removing the need for a separate validation set The AL cycle begins by finetuning the pretrained model on the seed set. If the size of the seed set is large enough, the seed set may be partitioned into a training set and a validation set, and early stopping may be performed on the validation set. However, in few-shot settings, labeling costs may be high, and the seed set may be too small to meaningfully partition.

Instead of an arbitrary fixed number of finetuning steps, we propose an alternative method to terminate finetuning in the absence of a validation set. Specifically, we find that a simple but effective heuristic is to stop finetuning when the training loss decreases to 0.1% of the original training loss at the start of finetuning. In our experiments, this heuristic performs as well as early stopping on an actual validation set (see Appendix D for more details). We also leverage standard learning rates and other hyperparameters recommended by model developers (see Appendix B).

By using a standardized recipe across tasks and removing the need for a separate validation set, our AL pipeline is more robust to the real-world difficulties of deploying AL where use of a validation set is impractical [37, 44], although further work is needed to capture the full extent of this recipe's generalizability.

3 Datasets

We consider a variety of datasets where task ambiguity manifests through a scarcity of particular kinds of examples. We consider two such kinds of examples: those defined by combinations of causal and spurious features (typical vs atypical backgrounds) as well as those defined by unseen attributes that shift during deployment (product categories and camera trap locations). These datasets provide an empirical testbed for the ability of pretrained models to choose disambiguating examples using active learning (AL).

3.1 Distinguishing causal from spurious features

Spurious correlations arise when multiple features are predictive of the label in a training dataset, yet it is ambiguous which ones are causally linked to the task label [21]. We consider two such datasets, and see whether AL can choose the *disambiguating examples* where the spurious features are not copresent with the causal features:

Waterbirds The Waterbirds dataset [48] consists of photographs of landbirds or waterbirds digitally edited onto land or water backgrounds. The task is to classify whether the bird is a landbird or a waterbird. In the train set, 77% of the pictures feature landbirds and 23% waterbirds. 95% of both landbirds and waterbirds appear on land and water backgrounds, respectively. In the validation and test sets, this percentage is decreased to 50%, instead. Thus, the image background is a spurious feature the model may come to rely on when making the prediction.

Treeperson As the Waterbirds dataset was synthetically generated, we also consider a dataset where we perform classification over real, unedited images with spuriously correlated objects. We use the object annotations in Visual Genome [32] to create a new dataset of 8,638 images called Treeperson, for which the task is to predict whether a person is in a given image. While 50% of the images contain a person in this dataset, each image also contains either a tree or a building, and the presence of these objects is spuriously correlated with the presence of people. At train time, 90% of training images with people contain a building, while 90% of training images without people contain a tree. Thus, a model may be incentivized to form representations that classify according to the presence of trees and buildings, rather than the presence of the actual causal variable of interest (people). These values are changed to 50% at test time, removing this correlation to evaluate how well the model learned the actual task of interest. For more details on this dataset, see Appendix C.

3.2 Measuring robustness to distribution shift

Distribution shifts occur when algorithms are evaluated on different data distributions than the ones they were trained on. Examples include changing the location or time of day that photos were taken, or changing the topic or author of a particular textual source. These shifts can reduce performance,

and we consider whether AL can help choose diverse, informative examples that clarify how the model should behave over a range of natural distribution shifts.

iWildCam2020-WILDS This dataset considers the task of species classification from a database of photos taken from wildlife camera traps [5, 30]. The dataset is unbalanced, with most images containing no animal, and the distribution of camera locations and species changes between the in domain (ID) and out-of-domain (OOD) subsets.

Amazon-WILDS This dataset considers the task of predicting the number of stars associated with the text of a given Amazon review [42, 30]. The reviewers are different in the training set versus the test set, and the task is to perform as well as possible on this set of new reviewers. In addition to the number of stars, we also consider model performance stratified by different product types, which highlights minority subgroups whose categorization is not visible to the model.

4 Experimental Setup

4.1 Models and Training

Vision For computer vision datasets, we finetune BiT [31], a recently-proposed family of vision models which have achieved state-of-the-art performance on several vision tasks. We primarily consider the BiT-M-R50x1 model, pretrained on ImageNet-21k [13]. To explore the effectiveness of larger architectures and pretraining sources, in Section 6.2 we also consider performance achieved by the same-size BiT-S-R50x1, trained on ImageNet-1k, and the deeper BiT-M-R101x1 model, also trained on ImageNet-21k. These models have been shown to have emergent few-shot learning abilities, where strong classifiers for new tasks can be obtained by simply finetuning on tens or hundreds of examples with typical gradient descent techniques (rather than meta-learning techniques, for example).

Text For the text dataset (Amazon), we use RoBERTa-Large [36], another pretrained model with similar properties as BiT, and a representative of the BERT [15] family of models which together have obtained state-of-the-art scores on modern NLP benchmarks [59].

Other details, including hyperparameters and seed set/acquisition sizes are deferred to Appendix B.

Random acquisition baseline As a running baseline, we compare to the same model finetuned with a *random acquisition* function (equivalent to not doing AL). That is, $a(x; \mathcal{M}) = \text{rand}(0, 1)$, so we simply sample a random batch of data from the pool at each acquisition step.

Comparison with non-pretrained models To examine whether effective AL is a result of the pretraining process, we also compare to the performance observed when applying AL to a *randomly initialized*, instead of pretrained, BiT-M-R50x1.

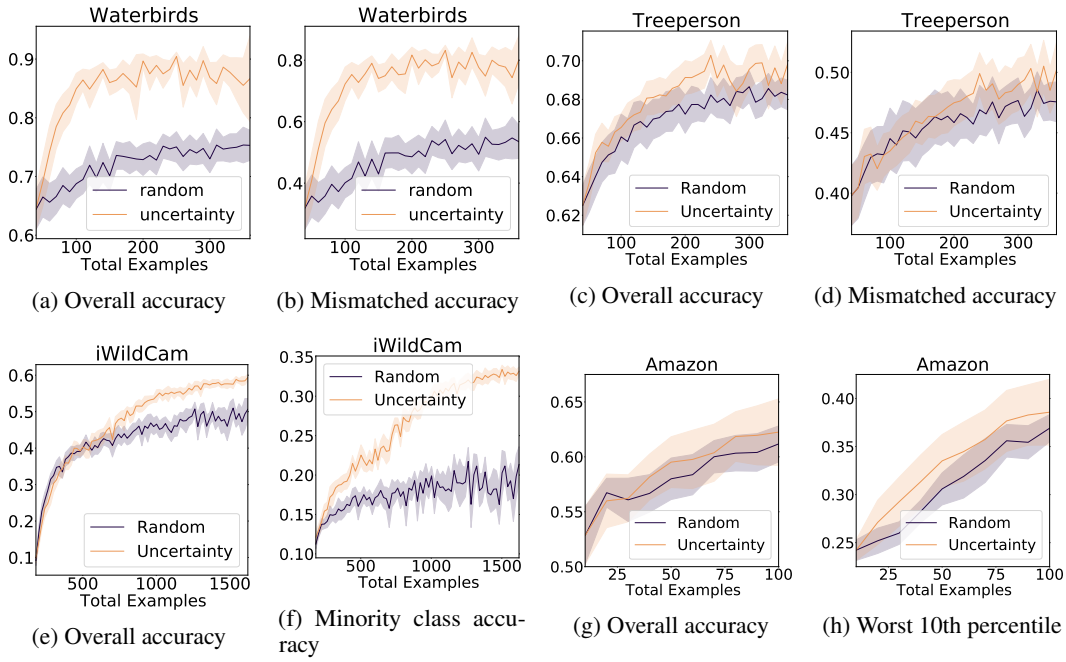
5 Results

5.1 Accuracy per acquisition

For a general measure of success, in Figure 2 we plot the accuracy of AL versus random sampling on the validation datasets as a function of the number of samples acquired during training.

Waterbirds Waterbirds is evaluated on a balanced dataset where the foreground and background are not correlated. In this setting, uncertainty sampling achieves a **+11% improvement** in average validation accuracy over random sampling (Figure 2a). This comes primarily from a **+25% average increase** across the landbird-on-water and waterbird-on-land images (i.e. those without the spurious correlation; Figure 2b). Uncertainty sampling required **5x fewer labels** than random sampling to achieve random sampling's final accuracy.

Amazon In the Amazon dataset, we also see gains from AL in the OOD setting, including **+1% on average** across reviewers, and **+2.5% on the worst 10th percentile** (Figure 2g). This suggests that



(a) Overall accuracy (b) Mismatched accuracy (c) Overall accuracy (d) Mismatched accuracy

(e) Overall accuracy (f) Minority class accuracy (g) Overall accuracy (h) Worst 10th percentile

Figure 2: **Uncertainty sampling outperforms random sampling on all datasets, especially on minority classes.** Class-balanced accuracies displayed for Figure 2f. Shaded regions represent 95% CIs (Gaussian approx.).

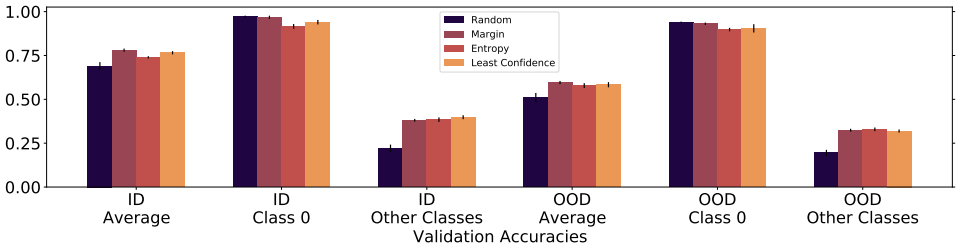


Figure 3: **All types of uncertainty sampling outperform random sampling on iWildCam.** Class 0 represents the majority class in iWildCam (no animal present).

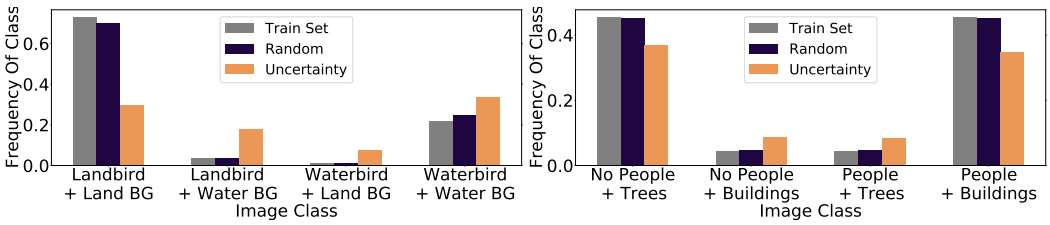
our AL recipe may be of use outside of BiT or vision settings more broadly. While the final difference between uncertainty and random sampling is not large, it is statistically significant. Uncertainty sampling required **1.3x fewer labels** than random sampling to achieve the same final accuracy.

iWildCam With the iWildCam dataset, uncertainty sampling achieved a **+9% improvement** upon random sampling. Uncertainty sampling also required **1.8x fewer labels** than random sampling to achieve random sampling's final accuracy (Figure 2e).

Treeperson In the Treeperson dataset, uncertainty sampling is **+2% improved** over random sampling by the end of training (Figure 2c). Uncertainty sampling required **1.6x fewer labels** than random sampling to achieve random sampling's final accuracy.

5.2 Additional AL methods

We consider two additional AL methods in addition to least confidence sampling: 1) entropy sampling, which chooses the example that maximizes the entropy of the model's predictive distribution, and 2) margin sampling, which chooses the example with the smallest difference between the first and second



(a) Waterbirds (b) Treeperson

Figure 4: **Uncertainty sampling identifies and upsamples disambiguating examples.** For both Waterbirds and Treeperson, uncertainty sampling selectively acquires examples where the spurious and core features disagree. Y-axis: frequency of class in acquisitions. Oversampling is visible for subgroups where uncertainty sampling acquires examples above random chance.

most probable classes [50, 51]. We run experiments with all methods on the 182-class iWildCam dataset.[2] All methods significantly outperform random sampling (Figure 3).

6 Analysis

6.1 AL with pretrained models selects examples that resolve task ambiguity

Overall, we attribute improved performance to pretrained models' ability to identify and preferentially sample examples that resolve task ambiguity in the data. For example, for Waterbirds and Treeperson, we see the model select examples with atypical background combinations, as one might intuitively hope. Similarly, for Amazon and iWildCam we see the model upsample rare types of examples, even when they are not explicitly marked in the data.

Waterbirds Figure 4a depicts the rate at which uncertainty sampling acquires examples of each subgroup compared to the expected rate at which random sampling would acquire examples from those same subgroup. Examples where the bird and background are mismatched are heavily oversampled. We emphasize that these minority examples are not simply members of the minority *class* (waterbirds). Instead, the model identifies and preferentially upsamples *disambiguating* examples where the spurious feature (background) and the causal feature (bird type) disagree.

Treeperson For Treeperson we see the same pattern as in Waterbirds: the model identifies and upsamples examples where only one of the spurious or causal features is present, despite the spurious feature being latent (Figure 4b).

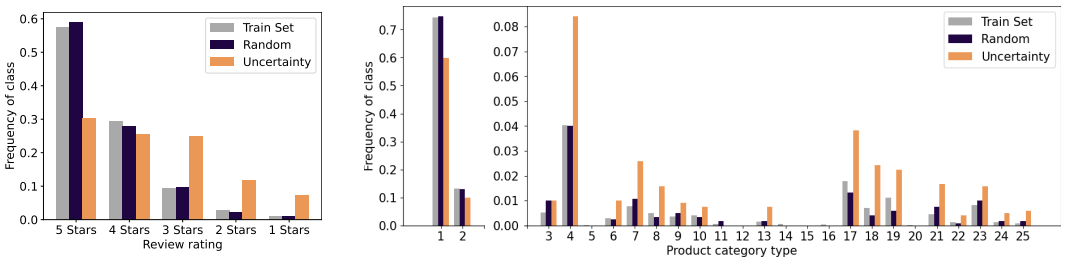
Amazon We also see similar behavior in the Amazon dataset, indicating our method's applicability to multiple modalities and pretrained models. Not only does the model upsample lower star ratings, which are less common, it can also upsample rarer product categories—a latent attribute (Figure 5).

6.2 Pretraining is the key ingredient in our experiments

What drives the success of AL in our experiments? We hypothesize that better AL is an *emergent property of the pretraining process*, and evaluate this hypothesis by comparing pretrained models to their corresponding non-pretrained counterparts. We also examine the effect of *model scale* on success at AL: if a model has been trained on more data (and has presumably learned to extract more semantically-relevant features), does this enable more effective AL?

Concretely, we conduct AL experiments with 3 pretrained models: BiT-S-R50x1, BiT-M-R50x1, BiT-M-R101x1, and their corresponding non-pretrained versions. We find that for our experiments with pretrained models, uncertainty acquisition outperformed random acquisition (Figures 6a, 6b and 6c). Importantly, AL on the non-pretrained models provided no or even negative benefit, even after exploring a range of different hyperparameter configurations. These controlled experiments

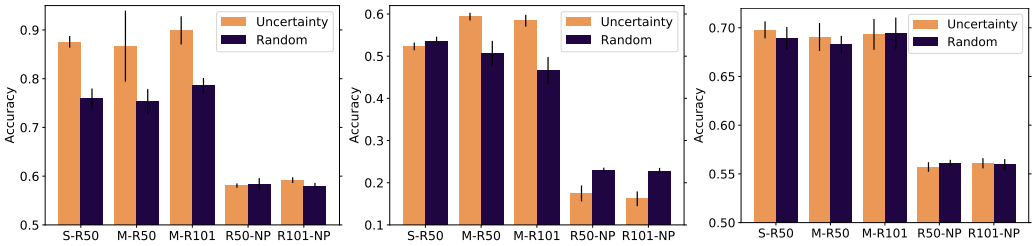
[2]Note that least confidence, entropy, and margin sampling are identical in the case of binary classification.



(a) Acquisitions by star rating

(b) Acquisitions by product category

Figure 5: **Uncertainty sampling upsamples both visible and latent minority subgroups.** Fraction of Amazon examples acquired by random and uncertainty sampling, stratified by star rating and product category. Upsampling is visible when the bar for uncertainty sampling is greater than the base prevalance in the unlabeled dataset available during training. Uncertainty sampling preferentially acquires examples with lower star ratings and rarer product categories, despite the latter attribute not being visible to the model. Note the separate y-axis for product categories 1 and 2 in (b).



(a) Waterbirds

(b) iWildCam

(c) Treeperson

Figure 6: **Uncertainty sampling only provides gains when using pretrained models.** S-R50, M-R50, and M-R101 correspond to the BiT-S-R50x1, BiT-M-R50x1, and BiT-M-R101x1 pretrained models, respectively, while R50-NP and R101NP correspond to ResNet models which are not pretrained. Error bars represent 95% CIs (Gaussian approx.).

provide strong evidence that pretraining is indeed crucial in our setting. That said, we do not claim that pretraining is the *only* way to enable good AL in settings of task ambiguity; other methods of addressing the so-called "cold start" problem (e.g. [20, 62]) may also prove fruitful or complementary.

Effect of scale In one case, we also see a demonstrable effect of *scale* on the efficacy of the AL process: the BiT-S-R50x1 model, which was pretrained on a smaller dataset than the BiT-M models (ImageNet-1k vs ImageNet-21k) fails to outperform random sampling on iWildCam, in contrast to the two other models pretrained on more data. This suggests a potential "scaling trend" for AL, where gains from AL may continue to grow as pretrained models are trained for longer on more data. However, we did not see a difference between BiT-M-50x1 and BiT-M-101x1, which were trained on the same dataset but have different numbers of parameters—suggesting that dataset size may be a more crucial variable than parameter count in some settings.

Impact of pretraining on acquisition patterns As an additional cross-check, we also observe that pretrained models acquire disambiguating subgroups much more efficiently than their non-pretrained counterparts. See Appendix G for additional figures and results.

6.3 Pretraining yields a better feature space for AL

While pretraining clearly improves the AL process, the mechanisms behind this improvement remain unclear. Given the strong theoretical results AL enjoys in the linear setting [2, 3, 40], we hypothesize that pretraining may aid AL by *linearizing* the features salient for task ambiguity. This hypothesis is further inspired by recent studies finding that a wide range of features are linearly separable in the feature spaces of large pretrained models [10].

	No Finetuning	Finetune On Seed Set (40)	Finetune On Seed Set (40) + 20
Average	0.402	0.42	0.424
Landbird /LandBG	0.389	0.416	0.42
Waterbird /LandBG	0.63	0.653	0.733
Landbird /WaterBG	0.426	0.467	0.447
Waterbird /WaterBG	0.435	0.418	0.419

(a) Group accuracies for linear classifier on Waterbirds image embeddings attained from a pretrained BiT model after various degrees of finetuning

	No Finetuning	Finetune On Seed Set (40)	Finetune On Seed Set (40) + 20
Average	0.311	0.32	0.34
Landbird /LandBG	0.306	0.321	0.369
Waterbird /LandBG	0.194	0.316	0.325
Landbird /WaterBG	0.286	0.248	0.262
Waterbird /WaterBG	0.337	0.33	0.255

(b) Group accuracies for linear classifier on Waterbirds image embeddings attained from an non-pretrained BiT model after various degrees of finetuning

Figure 7: **Both causal and spurious features are more linearly separable in pretrained models.**

To quantify this, we train linear classifiers on the second to last layer of the BiT models. The classifiers are trained to predict each image's bird type and background type (4 classes total, rebalanced to comprise 25% of the data). As shown in Figure 7, these classes are indeed far more linearly separable in pretrained models, providing evidence for this hypothesis.

We present additional investigations of t-SNE plots for pretrained and non-pretrained models in Appendix H, which demonstrates increased separation of latent classes for pretrained models, as well as how the acquired examples are closer to the class boundaries.

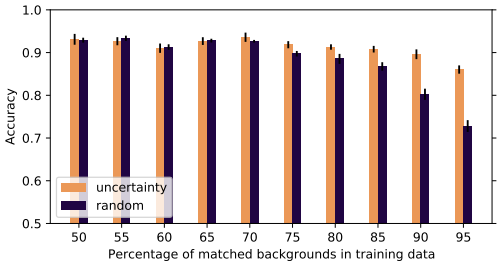


Figure 8: **Task ambiguity is the key factor driving the success of AL with pretrained models.** Accuracy on Waterbirds out-of-domain validation set for pretrained BiT-M models finetuned on datasets with different fractions of matched backgrounds. As disambiguating examples become more scarce, uncertainty sampling experiences far less of an accuracy drop vs random sampling.

6.4 How does the degree of task ambiguity impact AL?

Finally, we measure the impact of the *strength* of task ambiguity on AL. To do so, we construct variants of the Waterbirds dataset where the percentage of mismatched examples range from 95% to 50% but the marginal class probabilities remain fixed.[3] We then proceed with AL and report results in Section 6.3. We observe a clear trend where the gains from AL gradually increase as the percentage of mismatched examples increases to 95%. We find similarly clear trends in the upsampling ratio of mismatched backgrounds, shown in Figure 9. These results provides further evidence that task ambiguity is the key driving factor behind the success of AL in this setting.

6.5 Failure Cases

We also encounter some failure cases when training on datasets far from the distribution of the pretrained model. We perform preliminary experiments on Camelyon17-WILDS [4, 30], which considers tumor identification from tissue patches, and FMoW-WILDS [11, 30], which considers land-use classification from satellite images. AL performs comparably or worse than random sampling on these datasets, even when using a pretrained BiT model, suggesting that specialized models may be necessary to see gains in domains very different from ImageNet-21k.[4]

7 Related work

Task ambiguity and specification Several works address ambiguity or poor specification in machine learning problems. [57] describe the problem of "inductive ambiguity identification," and describe AL as a promising potential solution that has failed to see practical success. [12] describe the problem of *underspecification*, where high variance, instability, and poor model performance

[3] We construct these datasets using the code at `https://github.com/kohpangwei/group_DRO`.

[4] However, we note that Camelyon17-WILDS is also known to exhibit high variance across seeds (see `https://wilds.stanford.edu/get_started/`) which may also be a contributing factor.

result from training overparameterized models that are underconstrained by their training datasets. [21] describes how task ambiguity can arise when both desirable and undesirable features are predictive of the training labels, a problem which several works seek to better characterize and address [41, 48, 54, 49]. Finally, [18] address task ambiguity in few-shot settings via a probabilistic meta-learning algorithm, and perform an AL experiment in a 1D regression setting. We build on these works by demonstrating that simple uncertainty sampling with pretrained models can be an effective approach to the task ambiguity problem across a wide variety of high-dimensional classification settings—including when the sources of task ambiguity are not known.

Uncertainty and distribution shift In the face of these challenges, several works have tried to quantify how much pretrained models know about problems or their own uncertainty about them. [46] propose a question answering dataset with unanswerable questions, where a model must abstain rather than proceeding with an answer. Pretraining can also improve the calibration of model uncertainty [24] and pretrained features can be used for out-of-distribution detection [47, 61]—observations that align with our findings that uncertainty sampling can identify minority subgroups in datasets. A related stream of work seeks to identify high-confidence examples that are predicted incorrectly [1, 33]; by contrast, our focus is on improving model behavior across the full range of examples. Finally, our observation that upsampling latent minority groups results in better performance aligns well with [49, 27, 35], which explore various upweighting or upsampling strategies. Importantly, however, our approach does not require these groups to be known in advance.

AL and example selection Active learning (AL) [34, 52, 51, 26] is a well-studied field that investigates how machine learning algorithms might automatically select helpful additional data points to maximize their performance. Such strategies are especially helpful in imbalanced settings [17, 39] and have been fruitfully applied to deep models [19, 6], including pretrained models [62, 38, 53]. Past work has also considered AL for few-shot learning [60]. We extend these works by considering AL for resolving task ambiguity, showing that pretrained models successfully choose examples based on their high-level semantics, such as atypical backgrounds or rare latent attributes. Also in contrast to prior work, we investigate the *role of pretraining itself* by performing equivalent experiments with non-pretrained models, and providing a potential mechanism for the difference.

Pretrained models and their emergent properties Our work contributes to a broader literature on how pretraining enables new kinds of model capabilities [7, 55], especially those holding across multiple domains [56]. For example, [8] identify the phenomenon of in-context learning, where tasks can be specified for models through a language modeling prompt, while [9] discover that a self-supervised vision model implicitly learns high-quality segmentation maps visible through attention scores. [28, 25] conduct scaling laws experiments which chart how capabilities emerge with scale. We identify a new model capability that is significantly improved by pretraining: the capacity to actively learn and resolve task ambiguity in high-dimensional settings.

8 Discussion and Limitations

We show that pretrained models can preemptively resolve task ambiguity through active learning (AL), without requiring humans to anticipate these possible failure modes in advance. We find that AL helps across a variety of settings where data is spuriously correlated, undergoes domain shift, or contains unlabeled subgroups. These behaviors emerge most clearly as a result of large-scale pretraining, suggesting that AL may be an underappreciated tool for increasing the reliability of systems in real-world settings.

Of course, AL is no cure-all for resolving task ambiguity. First, it requires a human in the loop, which increases the time required to train a model compared to random sampling. Second, it requires the labeling method to be relatively free of noise—this may be acceptable if annotators are domain experts or are well-trained, but may also increase the cost per acquired example. Third, it is limited by the range of examples present in the unlabeled dataset—a model cannot elicit labels for examples that do not exist.

Finally, we note the opportunity for an exciting array of future work, including broader investigation of these methods across domains such as medical, scientific, or industrial settings, as well as better understanding how pretraining shapes AL as models continue to scale.

References

[1] Josh Attenberg, Panagiotis G. Ipeirotis, and Foster J. Provost. Beat the machine: Challenging workers to find the unknown unknowns. In *Human Computation*, 2011.

[2] Maria-Florina Balcan, Andrei Z. Broder, and Tong Zhang. Margin based active learning. In *COLT*, 2007.

[3] Maria-Florina Balcan and Philip M. Long. Active and passive learning of linear separators under log-concave distributions. *ArXiv*, abs/1211.1082, 2013.

[4] Péter Bándi, Oscar G. F. Geessink, Quirine F Manson, Marcory Crf van Dijk, Maschenka C. A. Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, Quanzheng Li, Farhad Ghazvinian Zanjani, Svitlana Zinger, Keisuke Fukuta, Daisuke Komura, Vlado Ovtcharov, Shenghua Cheng, Shaoqun Zeng, Jeppe Thagaard, Anders Bjorholm Dahl, Huangjing Lin, Hao Chen, Ludwig Jacobsson, Martin Hedlund, Melih Çetin, Eren Halıcı, Hunter Jackson, Richard Chen, Fabian Both, Jörg K.H. Franke, Heidi V. N. Küsters-Vandevelde, W. Vreuls, Peter Bult, Bram van Ginneken, Jeroen A. van der Laak, and Geert J. S. Litjens. From detection of individual metastases to classification of lymph node status at the patient level: The camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 38:550–560, 2019.

[5] Sara Beery, Elijah Cole, and Arvi Gjoka. The iwildcam 2020 competition dataset. *ArXiv*, abs/2004.10340, 2020.

[6] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

[7] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, D. Card, Rodrigo Castellon, Niladri S. Chatterji, Annie Chen, Kathleen Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, J.C. Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jackson K. Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *ArXiv*, abs/2108.07258, 2021.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *ArXiv*, abs/2104.14294, 2021.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

[11] Gordon A. Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6172–6180, 2018.

[12] Alexander D'Amour, Katherine A. Heller, Dan I. Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yi-An Ma, Cory Y. McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin G. Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. Underspecification presents challenges for credibility in modern machine learning. *ArXiv*, abs/2011.03395, 2020.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[14] Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In *EMNLP*, 2020.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021.

[17] Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 127–136, 2007.

[18] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *NeurIPS*, 2018.

[19] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192, 2017.

[20] Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö Arık, Larry S Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *European Conference on Computer Vision*, pages 510–526. Springer, 2020.

[21] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix Wichmann. Shortcut learning in deep neural networks. *ArXiv*, abs/2004.07780, 2020.

[22] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Christopher Olah. Multimodal neurons in artificial neural networks. 2021.

[23] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017.

[24] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *ICML*, 2019.

[25] T. J. Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. *ArXiv*, abs/2010.14701, 2020.

[26] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[27] Badr Youbi Idrissi, Martín Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. *ArXiv*, abs/2110.14503, 2021.

[28] Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv*, abs/2001.08361, 2020.

[29] Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher D. Manning. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. *ArXiv*, abs/2107.02331, 2021.

[30] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Wei hua Hu, Michihiro Yasunaga, Richard L. Phillips, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

[31] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020.

[32] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016.

[33] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. In *AAAI*, 2017.

[34] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994.

[35] Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *ICML*, 2021.

[36] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[37] David Lowell, Zachary Chase Lipton, and Byron C. Wallace. Practical obstacles to deploying active learning. In *EMNLP*, 2019.

[38] Katerina Margatina, Loïc Barrault, and Nikolaos Aletras. Bayesian active learning with pre-trained language models. *ArXiv*, abs/2104.08320, 2021.

[39] Stephen Mussmann, Robin Jia, and Percy Liang. On the importance of adaptive data collection for extremely imbalanced pairwise tasks. In *FINDINGS*, 2020.

[40] Stephen Mussmann and Percy Liang. On the relationship between data efficiency and error for uncertainty sampling. In *ICML*, 2018.

[41] Vaishnavh Nagarajan, Anders Johan Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. *ArXiv*, abs/2010.15775, 2021.

[42] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP*, 2019.

[43] Christopher Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Q. Ye, and A. Mordvintsev. The building blocks of interpretability. 2018.

[44] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. *ArXiv*, abs/2105.11447, 2021.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[46] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *ACL*, 2018.

[47] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *CVPR*, 2021.

[48] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *ArXiv*, abs/1911.08731, 2019.

[49] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. *ArXiv*, abs/2005.04345, 2020.

[50] Marten Scheffer, Stephen R. Carpenter, Jonathan A. Foley, Carl Folke, and Brian H. Walker. Catastrophic shifts in ecosystems. *Nature*, 413:591–596, 2001.

[51] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[52] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, 2008.

[53] Artem Shelmanov, Dmitri Puzyrev, Lyubov Kupriyanova, Denis I. Belyakov, Daniil Larionov, Nikita Khromov, Olga Kozlova, E. Artemova, Dmitry V. Dylov, and Alexander Panchenko. Active learning for sequence tagging with deep pre-trained models and bayesian uncertainty estimates. In *EACL*, 2021.

[54] Megha Srivastava, Tatsunori B. Hashimoto, and Percy Liang. Robustness to spurious correlations via human annotations. In *ICML*, 2020.

[55] Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. *ArXiv*, abs/2102.02503, 2021.

[56] Alex Tamkin, Vincent Liu, Rongfei Lu, Daniel E Fein, Colin Schultz, and Noah D. Goodman. Dabs: A domain-agnostic benchmark for self-supervised learning. *ArXiv*, abs/2111.12062, 2021.

[57] Jessica Taylor, Eliezer Yudkowsky, Patrick LaVictoire, and Andrew Critch. Alignment for advanced machine learning systems. 2020.

[58] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[59] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

[60] Mark P. Woodward and Chelsea Finn. Active one-shot learning. *ArXiv*, abs/1702.06559, 2017.

[61] Mike Wu and Noah D. Goodman. A simple framework for uncertainty in contrastive learning. *ArXiv*, abs/2010.02038, 2020.

[62] Michelle Yuan, Hsuan-Tien Lin, and Jordan L. Boyd-Graber. Cold-start active learning through self-supervised language modeling. In *EMNLP*, 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In Section 8
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In supplementary material
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Code will be released soon
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In supplementary material
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] License information will be added next revision.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Code release

Code and training scripts will be released soon.

B Additional experimental details

Here we provide some additional experimental details.

All BiT runs use the same default settings specified in the BiT paper [31] and the BiT GitHub repo: `https://github.com/google-research/big_transfer`

Settings that are constant across all BiT runs:

1. **Optimizer** - SGD with momentum 0.9
2. **Learning Rate** - Base learning rate 0.003. Linear warm up to this rate, then staircase decay. Exact schedule depends on dataset size, but for our few shot setting, this means: (1) linear warm up in the first 100 steps to 0.003, then (2) decay 10 fold every 100 steps; (3) after 500 steps, stop and move on to the next acquisition round.
3. **Data Augmentation** - Random cropping and flipping. See our `repo/utils/datasets/load` for details. Also available in BiT repo.
4. **Batch Size** - 32 when training, split into gradient accumulation microbatches of size 8.
5. **Early stopping condition** - When the training loss reaches below 0.001 times the original training loss.

Settings that varied between image datasets:

1. **Size of Initial Seed Set** - 40 for Waterbirds, 40 for Treeperson, 182 for iWildCam
2. **Size of Training Pool** - Entire training set for Waterbirds, Treeperson. A random subset of 12000 examples (re-drawn for each acquisition) from the entire training set of iWildCam.
3. **Number of Acquired Examples** - 320 for Waterbirds, 320 for Treeperson, 1456 for iWildCam.
4. **Number of Examples Acquired Each Acquisition** - 5 for Waterbirds, 5 for Treeperson, 20 for iWildCam.

Settings used for Amazon + RoBERTa-Large runs:

1. **Optimizer** - AdamW with default hyperparameters ($\beta_1 = 0.99, \beta_2 = 0.999$, weight decay = 0.1).
2. **Learning Rate** - LR = 1e-6.
3. **Batch Size** - 2 (due to memory considerations).
4. **Early Stopping Condition** - When the training loss reaches below 0.001 times the original training loss.
5. **Size of Initial Seed Set** - 10.
6. **Size of Training Pool** - A random subset of 2000 examples (re-drawn for each acquisition) from the entire training set.
7. **Number of Acquired Examples** - 90.
8. **Number of Examples Acquired Each Acquisition** - 2.

C Treeperson dataset

The Treeperson dataset is composed of images from Visual Genome [32] with different compositions of detected objects.

Training set composition by subclass:

- Person and Building: 3700
- Person and Tree: 370
- No Person and Building: 370
- No Person and Tree: 3700

The validation set contains 498 examples of each subclass.

The following annotated objects were used to form the different subclasses:

- Person: person, people, man, men, woman, women
- Building: building, buildings
- Tree: tree, trees, leaf, leaves, grass

The training set and validation set were drawn randomly from qualifying images in Visual Genome's training set and validation set, respectively.

D Early stopping condition

At the outset of this work, we explored if we could identify a heuristic for stopping training when there was no validation set present. We compared how the BiT model would perform if it stopped the finetuning step based off of when the validation accuracy plateaued versus when the training loss decayed to be 0.1% of its original value.

We ran a smaller experiment than the standard Waterbirds parameters we described in Appendix B. Namely, our seed set was of size 32, we acquired 64 examples on top of that, and we acquired one example at a time. We also defined the validation accuracy as having plateaued the fifth time it did not increase.

We ran twelve paired experiments where for twelve different randomized seed sets, we performed the Waterbirds experiment four times - with random sampling and stopping when validation accuracy plateaued, with random sampling and stopping when training loss decayed, with uncertainty sampling and stopping when validation accuracy plateaued, and with uncertainty sampling and stopping when training loss decayed.

We found that these experiments achieved:

1. Random sampling + Stop from validation accuracy: average accuracy = 69.11%, average duration = 3.5 hours
2. Random sampling + Stop from training loss: average accuracy = 69.93%, average duration = 38 minutes
3. Uncertainty sampling + Stop from validation accuracy: average accuracy = 83.64%, average duration = 4.5 hours
4. Uncertainty sampling + Stop from training loss: average accuracy = 85.62%, average duration = 74 minutes

Thus, we concluded that the by stopping our finetuning step just by waiting for the training loss to decay to 0.001 of its original values, we could achieve comparable if not better accuracies, spend a fraction of the time, and remove the need for a labeled validation set.

We did not consider other values besides 0.001 when conducting this search, though one could consider optimize this number in a fair way on a separate set of labeled datasets (in general one should not optimize AL pipelines using the validation sets they are being evaluated on).

E How the degree of task ambiguity impacts acquisitions

In Figure 9, we also plot the degree of upsampling that occurs for mismatched examples for different degrees of task ambiguity. As disambiguating examples become more scarce, uncertainty sampling increasingly upsamples them relative to their prevalence in the unlabeled training set.

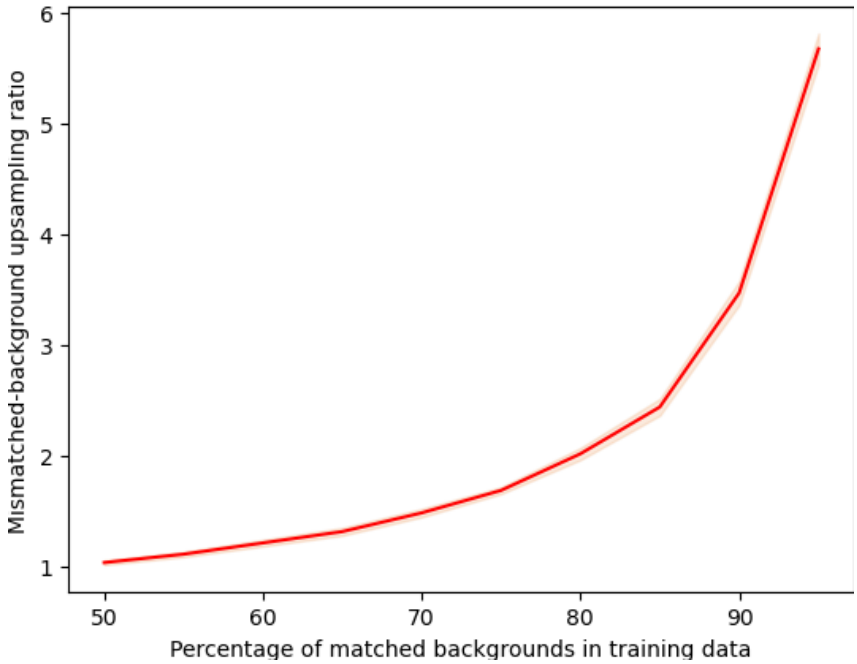
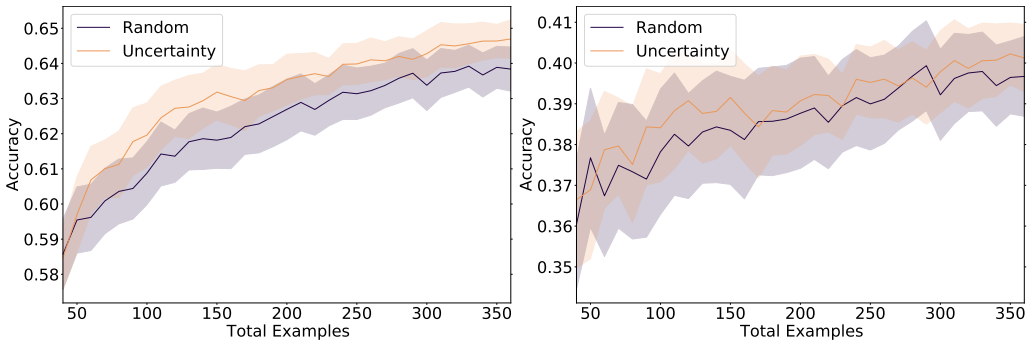


Figure 9: **Stronger task ambiguity results in more upsampling of disambiguating examples** Upsampling ratio of Waterbirds examples with mismatched backgrounds for pretrained BiT-M models finetuned on datasets with different fractions of matched backgrounds. As disambiguating examples become more scarce, uncertainty sampling increasingly upsamples them relative to their prevalence in the unlabeled training set.

F Vision transformer model

We broaden our coverage of computer vision models to include vision transformers [16], the other major architecture family currently in use. We train ViT-16/B,[5] on Treeperson. (Figure 10). This vision transformer was pretrained on ImageNet-21k for fewer epochs then BiT (9 vs 70), which is reflected by its lower oversampling of minority classes and comparatively smaller gains in accuracy compared to BiT (Figure 11).



(a) ViT (b) ViT

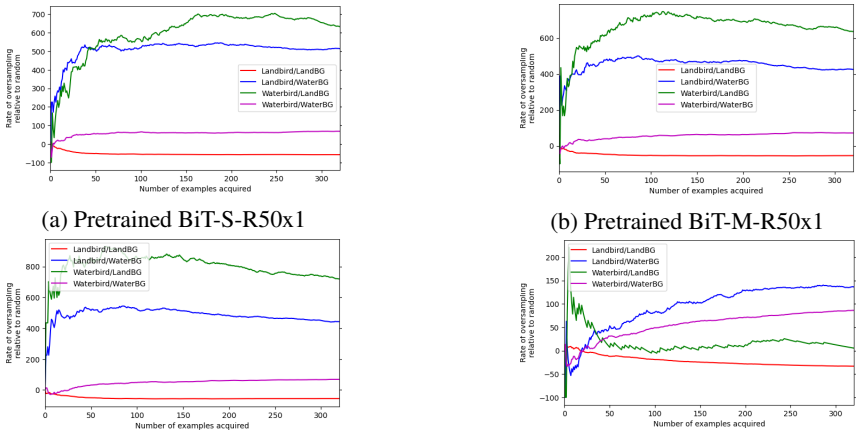
Figure 10: **AL improves upon random sampling for Treeperson when using ViT-B/16.** However, the gains are not as large as for BiT, which was pretrained for 8x more epochs.

[5]https://huggingface.co/google/vit-base-patch16-224

Figure 11: **Acquisitions for Treeperson using ViT.** The ViT model requested labels for minority classes at a significantly lower rate than did the BiT model. However, it still upsamples atypical backgrounds more than random sampling.

G Effect of model scaling and pretraining on acquisition

For the Waterbirds model scaling experiment, we track the examples each model acquires. These are presented in Figure 12. The acquisition patterns of all the pretrained models look fairly similar—they upsample both minority (landbird/water-background, waterbird/land-background) subclasses. However, the non-pretrained model is unable to capture that distinction, and is only able to upsample images with a water background, resulting in worse performance. A summary of final class acquisition ratios is available in Figure 13.



(a) Pretrained BiT-S-R50x1

(b) Pretrained BiT-M-R50x1

(c) Pretrained BiT-M-R101x1

(d) Non-pretrained ResNet R50x1

Figure 12: **All pretrained models acquire disambiguating subgroups much more efficiently than their non-pretrained counterparts on Waterbirds.** The pretrained models do not simply oversample based on the the bird or the background; instead they oversample disambiguating examples which have mismatched backgrounds. By contrast, the non-pretrained model only oversamples images with a water background, and accordingly is less able to perform well on the balanced validation set.

H Visualization Of Image Embeddings Of Pretrained Bit-M

t-SNE visualization We visualize the second-to-last layer embeddings of BiT (without any finetuning) using t-SNE [58].

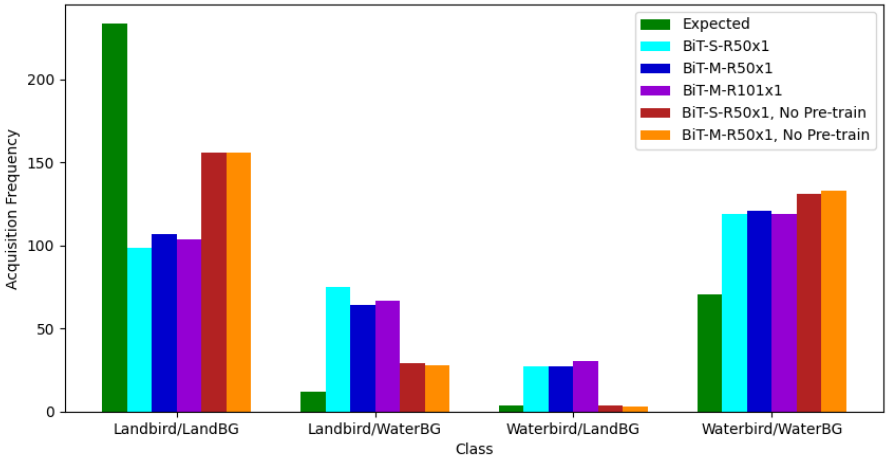
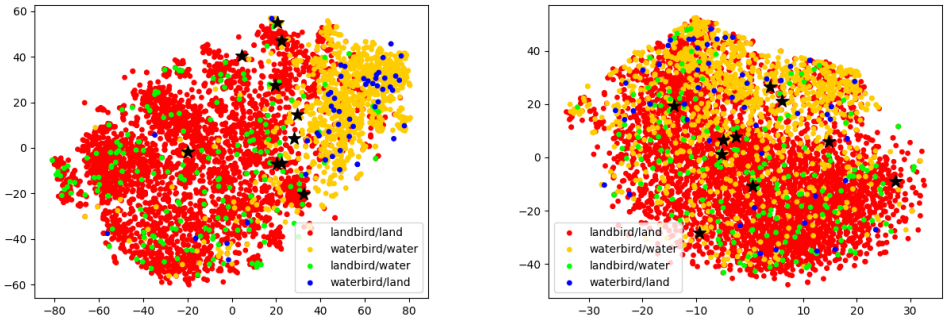


Figure 13: **Pretrained models ask for labels of images with mismatched backgrounds, while non-pretrained models do not.**

As seen in Figure 14b, when performing t-SNE on the Waterbirds image embeddings, we find that the t-SNE of the pretrained model has much more structure than that of the non-pretrained model. In the non-pretrained model, the image embeddings of the landbird/land background, landbird/water background, and waterbird/water background classes are distributed uniformly about the center of the t-SNE projection. However, we observe much more structure in the pretrained model's t-SNE. In particular, the most noticible difference is that the landbird/water background and waterbird/land background classes can be found near the other landbird and waterbird images, respectively. This shows that even without any finetuning, the pretrained model already has learned features to help process the type of bird that appears in the image.

In the t-SNE plots, the black stars represent the embeddings of the first 10 images that were acquired by each model. For the pretrained model, the acquired examples very clearly fall along the waterbird-landbird boundary in the projected feature space. However, in the non-pretrained model, the acquired examples are distributed randomly. This suggests that the features acquired during pretraining enables them to select examples that fall near the decision boundary of a new unseen task.



(a) Pretrained BiT-M

(b) Non-pretrained BiT-M

Figure 14: **Without any finetuning, pretrained models already embed images into a useful feature space.** t-SNE of Waterbirds image embeddings from pretrained and non-pretrained BiT-M. Black stars denote acquired examples, which fall more consistently along the decision boundary between landbirds and waterbirds for pretrained models vs non-pretrained ones.

Potential Societal Impact Our work concerns more general conceptual understanding and use of pretrained models, making it hard to speculate directly about downstream societal impacts. In general, such methods are likely to amplify the existing uses of such models. For example, methods for actively aligning models to human intent could have positive or negative consequences depending on whether the intent was beneficent or malevolent. This highlights the need for careful design of societally-beneficial governance structures when deploying these models.

Compute used We used Titan Xp GPUs in an internal cluster. Each trial took on the order of hours.