Comparing Importance Sampling Based Methods for Mitigating the Effect of Class Imbalance

Indu Panigrahi and Richard Zhu
Princeton University, Princeton, NJ, USA
{indup, ryzhu}@princeton.edu

Abstract

Most state-of-the-art computer vision models heavily depend on data. However, many datasets exhibit extreme class imbalance which has been shown to negatively impact model performance (3; 24; 49). Among the training-time and data-generation solutions that have been explored (8; 32; 53; 9; 29), one subset that leverages existing data is *importance sampling* (5). A good deal of this work focuses primarily on the CIFAR-10 and CIFAR-100 datasets (53; 9) which fail to be representative of the scale, composition, and complexity of current state-of-the-art datasets. In this work, we explore and compare three techniques that derive from importance sampling: loss reweighting, undersampling, and oversampling. Specifically, we compare the effect of these techniques on the performance of two encoders on an impactful satellite imagery dataset, Planet's Amazon Rainforest dataset, in preparation for another work. Furthermore, we perform supplemental experimentation on a scene classification dataset, ADE20K, to test on a contrasting domain and clarify our results. Across both types of encoders, we find that upweighting the loss for and undersampling has a negigible effect on the performance on underrepresented classes. Additionally, our results suggest oversampling generally improves performance for the same underrepresented classes. Interestingly, our findings also indicate that there may exist some redundancy in data in the Planet dataset. Our work aims to provide a foundation for further work on the Planet dataset and similar domain-specific datasets. We open-source our code at https://github.com/RichardZhu123/514-class-imbalance.

1 Introduction

Detecting illegal slash-and-burn and mining activities in protected rainforests is crucial to ensuring the well-being of such areas, and satellite imagery can facilitate this process. As a result, domain experts need classification models to accurately detect such activities. However, the Planet Rainforest dataset (12; 45), like many large and widely-used datasets, exhibits a long-tail distributions where some classes have significantly more examples than others (Figure 1) (66; 35; 15; 68; 67; 18). This trend becomes more difficult to remediate when each image has multiple labels, as is the case with the Planet Rainforest dataset. In fact, the activities that we want to detect are some of the worst-represented classes in the dataset (Figure 1).

Oftentimes, models achieve higher accuracies on overrepresented classes than on underrepresented classes, similar to how a human would be better at classifying images corresponding to classes for which they have seen more examples (3; 24; 49; 20). However, this induced bias can lead to dangerous and unfair model predictions (4; 27; 56; 57). As machine learning research heads towards a direction of minimizing bias, training data plays an ever more important role in shaping the representations formed through learned parameters.

From a theoretical machine learning perspective, in addition to bounding generalization error as we evaluate models on test sets, we may also be interested in developing empirical methods for

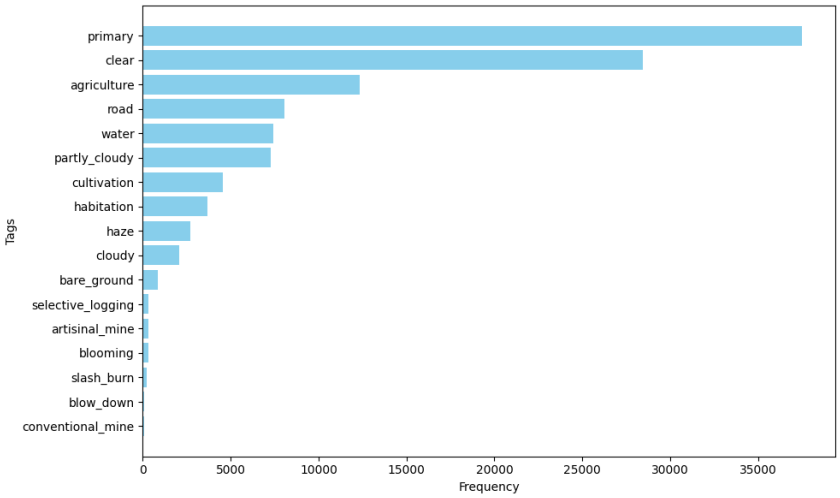


Figure 1: **Long-tail distribution of classes in the Planet dataset.** Important labels, such as "slash_and_burn" are some of the most underrepresented classes in the dataset.

minimizing expected error. By understanding the causes of poor generalization in practice, we can gain intuition in order to tighten the bounds learned in class when developing models in practice.

A natural solution to this issue of class imbalance would be to simply collect or generate more data. However, importance sampling covers a subset of methods that leverage existing data to mitigate the effects of class imbalance (5; 19; 25; 46; 28). To that end, we explore and compare three importance sampling based methods: loss reweighting, undersampling, and oversampling.

Specifically, we experiment with two datasets from different domains, the Planet Rainforest (12; 45) and the ADE20K (67) datasets, and with two image encoders, CLIP (42) and ResNet-18 (16). We primarily focus on the Planet dataset in preparation for another work, so our goals are slightly specific for the purposes of this course project. However, we include supplemental experimentation on ADE20K.

Across the two types of encoders, we find that up-weighting the loss on underrepresented classes has a negligible effect on performance. Additionally, we find that oversampling appears to be more effective than undersampling. That being said, our results suggest that undersampling could still be a valuable method if we mindfully select samples rather than at random. Lastly, our experiments indicate that there may exist some redundancy in the Planet dataset which motivates new avenues for future work.

2 Related Work

Class Imbalance Problem. Class imbalance occurs when some classes have significantly more examples than others in a dataset. Imbalanced distributions naturally arise across many types of datasets ranging from medical imaging to environmental satellite imagery (30; 44; 60; 17). While other forms of imbalances exist (e.g., object-level for object detection and pixel-level for segmentation), our work explores solutions for class imbalance since we examine image classification. While the class imbalance problem has been well-studied in traditional non-deep machine learning and computer vision, recent studies have noted the lack of work on studying the effects of and methods for counteracting class imbalance across a variety of datasets and imbalance levels (24). We work with a satellite imagery and a scene classification dataset to evaluate on different domains. Prior work has shown class imbalance to deteriorate model performance on underrepresented classes (55; 36; 13; 41). However, these classes can be extremely important, as is the case with the Planet dataset (Figure 1).

Data generation. A natural solution to the class imbalance problem would be to collect more data to balance the dataset. Since manual data collection can be quite arduous for specialized applications (e.g., medicine, botany, and geosciences) (2; 50; 58; 18), an alternative is to generate additional data via self-supervised learning (63; 22), semi-supervised learning (32; 61), or generative adversarial

networks (GANs) (8; 14). However, one common limitation is that the generated data can sometimes be low in quality, particularly for specialized applications such as for those represented in this paper (e.g. satellite imagery of rainforests). Furthermore, leveraging existing data would still be data-efficient but simpler. As a result, we choose to focus on comparing methods that leverage existing data.

Cost-sensitive learning. One such method is cost-sensitive learning. Cost-sensitive learning is a progressive training method where higher-cost examples contribute more significantly in updating the model. In practice, this means the user can tailor the model to heavily penalize certain misclassifications (10; 62). For example, if the model performs poorly on an underrepresented class, the user can increase the penalty for mistakes on that class by influencing the cost (i.e., loss). In this work, we implement a loss reweighting scheme which we describe in Section 3.

Importance Sampling. Another related subset of methods that overlaps with cost-sensitive learning is importance sampling. Importance sampling is useful when the test distribution q is not accurately reflected in the training distribution p (19; 25; 46; 28). For some example x, if we are estimating some function $f(x)$, importance sampling weights x based on its likelihood ratio $\frac{q(x)}{p(x)}$ as shown in Equation 1 (5).

$$\mathbb{E}_p\left[\frac{q(x)}{p(x)}f(x)\right] = \int_x \frac{q(x)}{p(x)}f(x)p(x)dx \tag{1}$$

$$= \int_x q(x)f(x)dx \tag{2}$$

$$= \mathbb{E}_q[f(x)] \tag{3}$$

The idea of importance sampling forms the basis for the three methods that we explore which we describe in the next section.

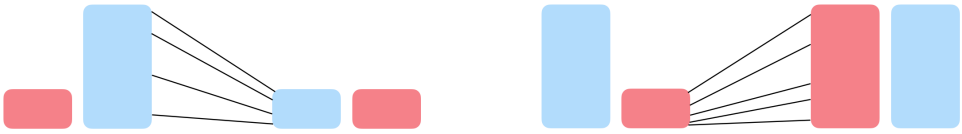
3 Methods

Our work explores and compares loss reweighting and two types of resampling: undersampling and oversampling. In this section, we provide a description of each method.

Reweighting loss. This technique reweights the loss at training time to amplify the loss for low-frequency classes and/or diminish the loss for other classes. One approach scales the loss of all low-frequency classes by a constant, while another scales the loss of all classes relative to their frequency. We choose to do the former because we find that it offers more granularity in controlling which classes are amplified in situations where two labels may be very similar. For example, if we have an image classification task where the classes "roadway" and "highway" appear moderately frequently, we may still wish to consider these classes as low-frequency classes to improve the model's ability to distinguish between the two (likely very similar) images.

Prior work has previously implemented variants of the latter approach. One such work evaluates the effectiveness of learning a mapping from classes to reweighting factors (53). Using an MLP with a single hidden layer, they observe some performance gains on the CIFAR-10 and -100 datasets, achieving test accuracies on ResNet-32 that are greater than or equal to those on a class-balanced version of the dataset. Another work simply scales the loss by the inverse class frequency. Their method, `class-balanced focal loss`, achieves a nominally better training loss on the `ILSVRC 2012` (47) and `iNaturalist 2018` (18) datasets after sufficiently many epochs, while results at earlier epochs underperform the default softmax cross-entropy (9). On CIFAR-10 and -100, they report gains, albeit marginal ones. We include a discussion about another related work that takes the latter approach in the context of our results for completeness (Section 5) (5).

Undersampling. Undersampling reduces the number of examples in overrepresented classes by randomly sampling a subset of a predefined size (Figure 2a). This size is usually the size of the lowest-frequency class (3; 34). This method effectively decreases the total number of elements. As a result, the number of examples seen from each class during training is generally more uniform, so intuitively, the model is expected to perform more uniformly across classes. One of the first papers



(a) **Undersampling.** (b) **Oversampling.**

Figure 2: **Types of Resampling.** Oversampling (b) trains on repeated elements from underrepresented classes, while undersampling (a) does the opposite, training on a subset of elements from overrepresented classes.

that discussed class imbalance highlighted the potential performance improvements of undersampling methods (29).

Since undersampling removes examples from the training set, this method appears to be more suitable for large datasets where we can afford to prune the dataset without significantly affecting results (3). While the exact threshold for such a decision seems nebulous, most prior work finds that the thresholds that we use (typically undersampling all elements to a threshold approximately equal to the number of elements in the least populous class being trained on), does not significantly decrease performance (3). Thus, applying undersampling to the Planet dataset seems reasonable.

Oversampling. Similar to undersampling, oversampling aims to create a more uniform distribution of training examples across classes. However, this process instead repeats training examples from low-frequency classes via random sampling until those classes reach a comparable frequency to the other classes (Figure 2b). Both undersampling and oversampling can be tricky for multi-label classification (e.g. with the Planet dataset). There exist several approaches to ensure a roughly similar number of elements in both resampling methods, and we discuss our approach in the next section (Section 4).

4 Experimental Details

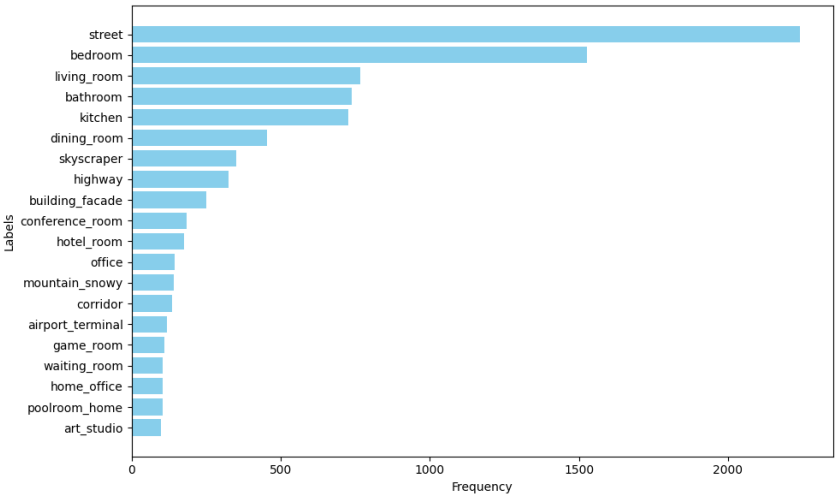
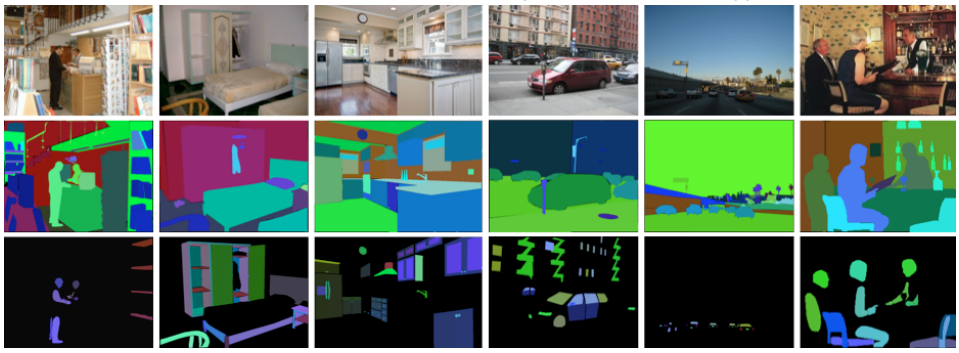


Figure 3: **Distribution of classes in ADE20K.**

Data and Tasks. We primarily experiment with the Planet Rainforest dataset (Figure 4a) and include supplemental experimentation on the ADE20K dataset (Figure 4b). The ADE20K dataset is a scene classification dataset, and the Planet dataset contains satellite imagery of the Amazon rainforest. We choose the Planet dataset in preparation for another work and consequently test multi-label classification. We include ADE20K in our reweighting experiments to disentangle the

results (Section 5), provide a contrasting domain, and test single-label classification. However, the primary focus of this work is on the Planet dataset.

(a) **Sample Images from the Planet dataset.** Figure from Planet's Kaggle website (12).



(b) **Sample Images from the ADE20K dataset.** ADE20K includes segmentation labels as shown; however, we only use the image classification labels. Figure from the ADE20K website (67).

Figure 4: **Sample images from the datasets used in this work.**

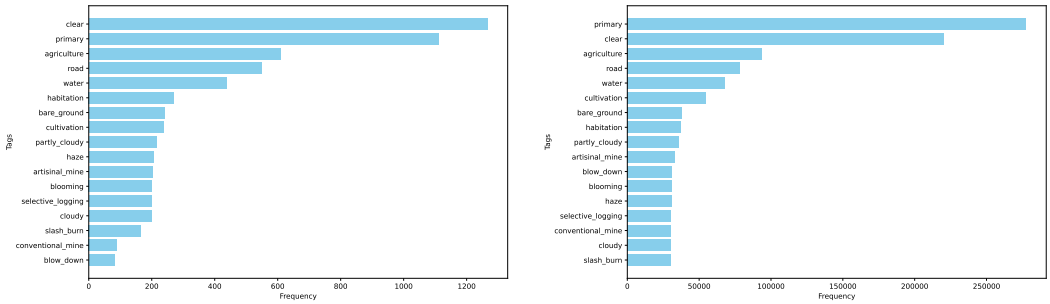
We use all 17 classes from Planet and select the 20 most frequent classes (out of the 150 total) from ADE20K. Both datasets exhibit long-tail distributions with Planet having the more severe of the two (Figure 1, 3). Since these are challenge datasets, the test sets are not publicly available, so we apply a 60-20-20% split to the training and validation sets combined to produce our training, validation, and test sets. In our results, we report our testing accuracies.

Define low-frequency classes. Prior works mostly tailor the definition of low-frequency classes to their datasets by selecting classes based on certain features or by predefining a frequency threshold (23; 40). A more general option is to define the low frequency classes to be those that fall one standard deviation below the mean. However, for Planet, the mean frequency is 5464.82, and the standard deviation is 8214.73 which is too large to follow this standard. Instead, we define the low-frequency classes to be those that have frequencies that fall within the 25th percentile of all the frequencies. To maintain consistency, we use the same definition to select the low-frequency classes in ADE20K.

Resampling. For undersampling, we choose a threshold of 200 for Planet because this is approximately the frequency of the lower-frequency classes. Thus, we randomly sample from classes with frequencies that exceed this threshold to reduce those classes to the threshold.

For oversampling, we choose a threshold of $30,000$ for Planet because this is approximately the frequency of the higher-frequency classes. Similar to undersampling, we randomly sample from classes with frequencies lower than this threshold until they reach the threshold. However, before doing so, we ensure that all the images from those classes are included so that each image appears at least once.

Since the images in Planet can have multiple labels, ensuring a perfectly uniform distribution is unreasonable. In particular, the "primary" label indicates the presence of trees, and "clear" indicates that there exists no cloud cover. As a result, these two labels overlap with many images, especially as



(a) **Undersampled distribution of the Planet dataset.** (b) **Oversampled distribution of the Planet dataset.**

Figure 5: **Resampled distributions of the Planet dataset.**

each image has a cloud cover label (i.e., "clear", "partly_cloudy", "haze"). However, we achieve a generally uniform distribution among the more disjoint classes (Figure 5) which is significantly less skewed than the original distribution (Figure 1).

Architecture. Our architecture consists of a pretrained encoder followed by a simple classifier. We experiment with two encoders pretrained on ImageNet: CLIP (42) and ResNet-18 (16). We implement a simple classifier in PyTorch that consists of the following layers: Linear, ReLU, Linear, and Sigmoid. We use a binary cross-entropy loss function. The encoders are frozen, so we specifically train the classifier.

Training. After performing a hyperparameter search, we decide to train each model for 20 epochs using a learning rate of $1e-3$, batch size of 64, and dropout rate of 0.2. We train all the models with the same hyperparameters to control for variance in performance caused by hyperparameters as do prior works.

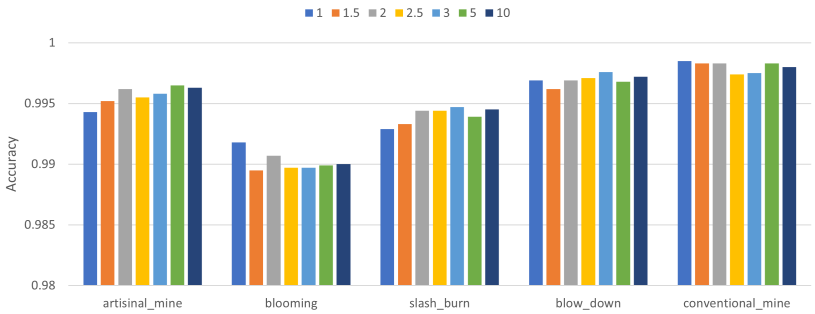
5 Results & Discussion

5.1 Up-weighting the loss for low-frequency classes does not have a significant effect on the performance on any of the classes.

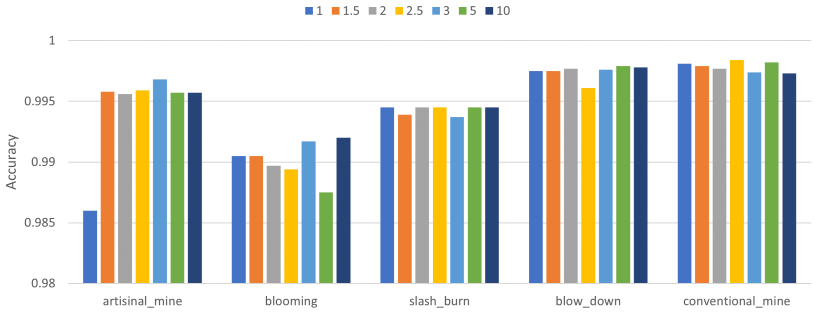
Across both models, we find that the performance does not exhibit a consistent improvement or decay when we up-weight the loss of the low-frequency classes in Planet (Figure 6, 7). The results for multi-label classification should be interpreted with the caveat that each image can have more than one label. For example, if an image has the labels "primary" and "slash_and_burn", then we would up-weight the loss on the image because "slash_and_burn" is a low-frequency class. However, in doing so, we implicitly up-weight the loss on "primary". Thus, the up-weighting is not cleanly done on one class and can cause sporadic drops or gains in performance. To disentangle these effects, we check the performance of the same up-weighting scheme on single-label classification. However, we again find that the performance of the models does not significantly change when we up-weight the loss of the low-frequency classes (Figure 8).

For single-label classification, the performance on an up-weighted class sometimes drops surprisingly (e.g., with the "game_room" label in Figure 8). However, a plausible explanation is that the training examples for that label do not reflect the testing distribution, and the model learns meaningless artifacts and/or makes a faulty association (38; 31; 37; 11). For example, if the training examples for the class "game_room" tend to contain pool tables and if the testing examples for "game_room" do not contain pool tables, the model could develop a dependence on the presence of pool tables to identify a scene as "game_room" and consequently misclassify the testing examples. This hypothesis could be investigated further by comparing the saliency maps generated for the training images to those generated for the testing images (51; 54; 52).

In addition to our experiments, this finding is partly supported by prior experimentation in Byrd & Lipton's work on importance sampling (5). Byrd & Lipton use a reweighting scheme that weights each class according to its relative frequency in the dataset. This scheme is slightly different from our setup in that we use a constant scaling factor for all classes, regardless of frequency. Nonetheless,



(a) **Up-weighting accuracy with CLIP encoder.**



(b) **Up-weighting accuracy with ResNet-18 encoder.**

Figure 6: **Up-weighting accuracy on the Planet dataset on low-frequency classes.** These graphs depict the accuracies when scaling the loss for low-frequency classes by factors of $1.5, 2, 2.5, 3, 5,$ and 10 with different encoders; the factor of 1 corresponds to the baseline accuracy. Both panels depict the accuracies on the low-frequency classes only; see Figure 7 for the remaining classes. For both encoders, we find that there exists no consistent trend across scaling factors or within a factor.

Byrd & Lipton found that the impact of reweighting the loss diminished over time, and our results are consistent with this finding.

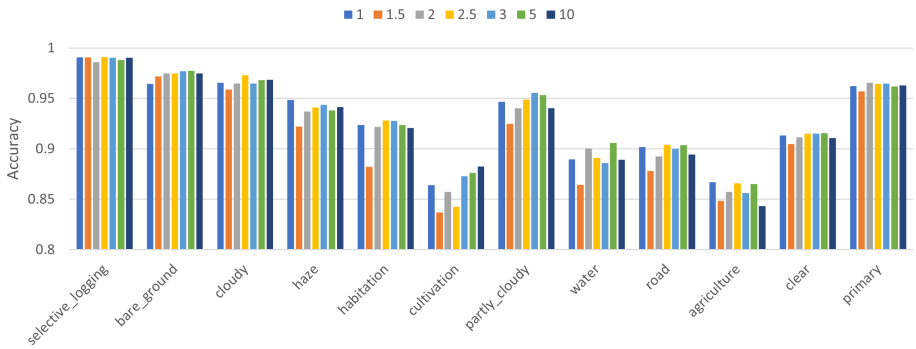
5.2 Undersampling produces a comparable performance to the baseline on the low-frequency classes while reducing the performance on the other classes.

Across both models, the performance when using undersampling roughly matches or surpasses the baseline performance on low-frequency classes (Figure 9). The situations where the performance improves compared to the baseline performance are surprising because we use the same images for these classes. One potential cause is stochasticity of training which could be resolved by instead training multiple undersampling and baseline models and averaging the performances assuming access to sufficient computational and storage resources.

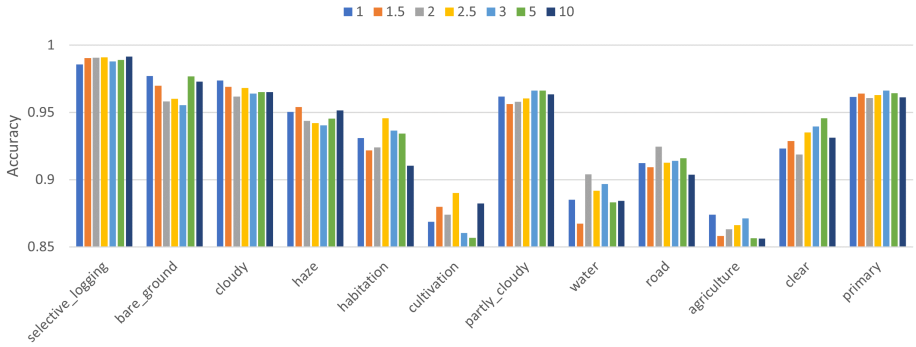
The performance on the remaining classes often drops (e.g., as with the "agriculture" and "partly_cloudy" labels) (Figure 10). This trend seems reasonable since undersampling removes data from these classes. However, it is interesting to note that the performance on some of these classes remains roughly the same (e.g., the "selective_logging" and "bare_ground" labels). This trend suggests that the examples removed from those classes were redundant with the examples that remained and that the examples that were removed from the negatively-impacted classes did not contain redundant information (59; 39; 1; 26). Furthermore, we use approximately 21 times less data when undersampling, so the fact that the performance is still reasonable across all classes is impressive and further supports the idea that there may exist some redundancy in the data.

5.3 Oversampling generally improves performance across low-frequency classes.

Among the low-frequency classes, oversampling generally matches or improves upon the baseline performance (Figure 9). Since oversampling artificially adds data by sampling from existing data,



(a) **Up-weighting accuracy on the remaining classes in Planet when using CLIP.**



(b) **Up-weighting accuracy on the remaining classes in Planet when using ResNet-18.**

Figure 7: **Up-weighting accuracy on Planet on remaining classes.** These graphs depict the accuracies on the remaining classes in Planet when scaling the loss for low-frequency classes by factors of $1.5, 2, 2.5, 3, 5$, and 10; the factor of 1 corresponds to the baseline accuracy.

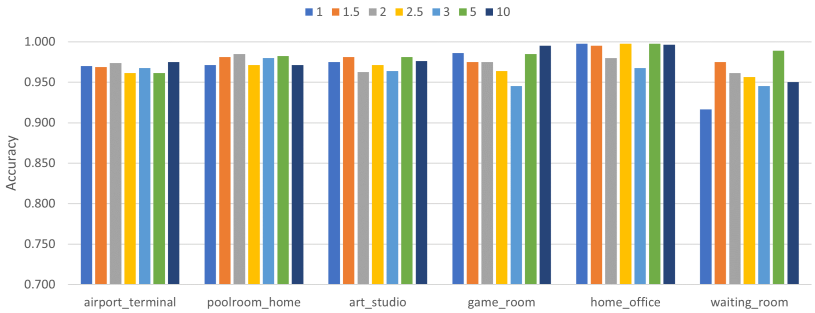
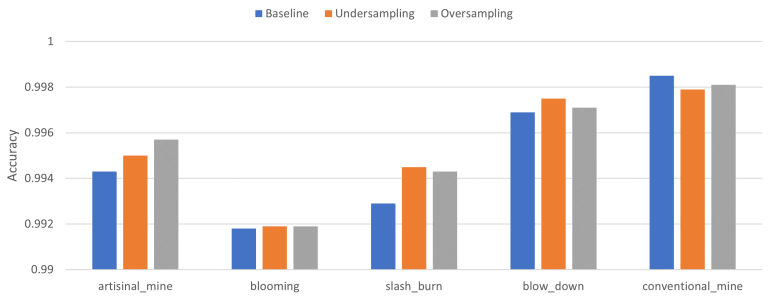
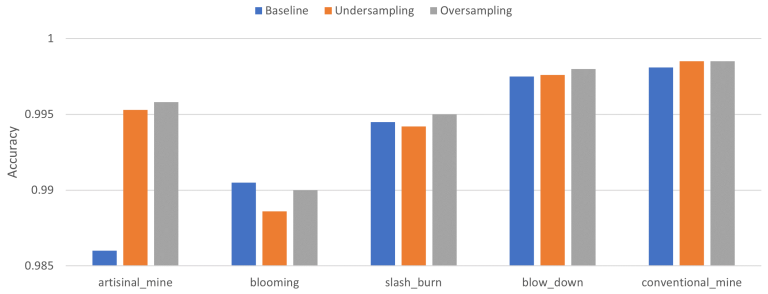


Figure 8: **Up-weighting accuracy on ADE20K with CLIP encoder on low-frequency classes.** This graph depicts the accuracies when using a CLIP encoder and scaling the loss for low-frequency classes by factors of $1.5, 2, 2.5, 3, 5$, and 10. The factor of 1 corresponds to the baseline accuracy. Similar to the results from the Planet dataset, we find that there is no consistent trend across the scaling factors. We find a similar trend on the remaining classes whose results can be found in the Appendix.

marginal improvements are reasonable to expect. For situations where the performance is matched (e.g., "blooming" and "blow_down" in Figure 9), the repetition of the data could have outweighed any improvement. In particular, it is interesting to note that, with ResNet-18, as the classes reduce in frequency (progress along the x-axis), the gap between the baseline and oversampling performance reduces (Figure 9b). This trend likely occurs because the fewer examples a class contains, the more frequently those examples will be seen during training with oversampling, so the model learns highly repetitive information about that class.

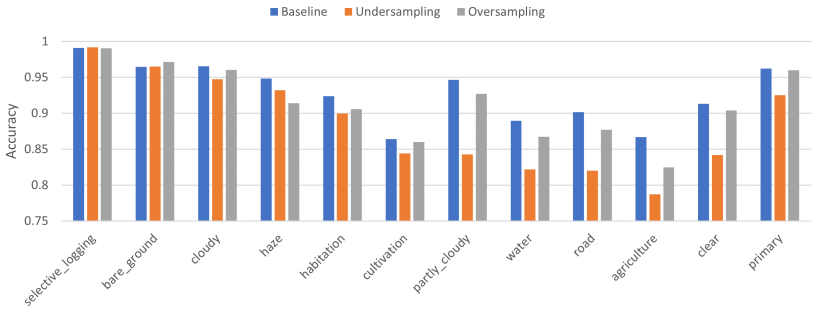


(a) **Resampling accuracy on Planet's low-frequency classes with CLIP encoder.**

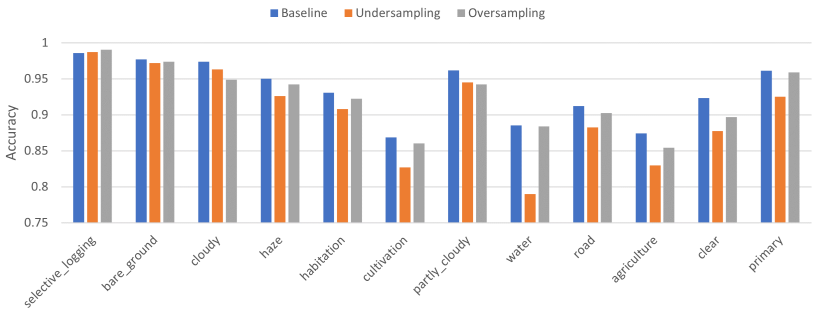


(b) **Resampling accuracy on Planet's low-frequency classes with ResNet-18 encoder.**

Figure 9: **Resampling accuracy on the low frequency classes in the Planet dataset.** The classes are sorted in descending order by frequency along the x-axis.



(a) **Resampling accuracy on Planet's remaining classes with CLIP encoder.**



(b) **Resampling accuracy on Planet's remaining classes with ResNet-18 encoder.**

Figure 10: **Resampling accuracy on the remaining classes in the Planet dataset.** The classes are sorted in ascending order by frequency along the x-axis.

Surprisingly, oversampling sometimes leads to a drop in performance in the remaining classes, though to a lesser degree than undersampling (Figure 10). This trend likely occurs because we stopped

training prematurely. Oversampling uses approximately 8 times more data than the baseline, so using the same number of epochs when training would not be sufficient. However, as noted in Section 4, we train all models under the same hyperparameters to control for comparison.

5.4 Comparison between methods

Surprisingly, up-weighting the loss on low-frequency classes, even by a relatively large factor of 10, does not produce consistent effects on the performance of any of the classes. In fact, the two resampling methods seem to produce more consistent effects than reweighting. Compared to undersampling, oversampling appears to have the potential to cause the most improvement provided more training time to account for the significant increase in data. However, the performance when undersampling is surprisingly high considering that we use 8 times less data from Planet and perhaps selecting data according to some criteria rather than at random would help.

6 Conclusion

In this work, we present a comparison of methods for mitigating the effect of class imbalance that are based on the idea of importance sampling. Specifically, we focus on a satellite imagery dataset and provide supplemental experimentation with a scene classification dataset. Across two types of encoders, CLIP and ResNet-18, we find that up-weighting the loss on low-frequency classes has a negligible effect on the performance. Additionally, we find that oversampling appears to be more effective than undersampling. However, we also find that undersampling could still be a valuable method though perhaps users should not select data at random.

In our discussion, we often noted that there likely exists some redundancy in the data. This idea is part of the motivation for data distillation which attempts to reduce a dataset to a few representative examples per class (43; 6; 48). Thus, this could be an interesting alternative to undersampling via random selection.

Additionally, we focus on bias due to underrepresentation. However, in our discussion, we also hint about bias due to spurious correlation (i.e., when the model forms a faulty association between certain image features and a class) (57). Thus, there exist other forms of bias to consider to develop truly robust models.

A macroscopic implication of these results is that, as a field, our understanding of how black-box deep neural networks operate is still nascent. As shown in this report, some of our intuitions on what should improve model performance (e.g., "more data should improve model performance" and "up-weighting loss should improve performance") do not necessarily hold. As machine learning rapidly expands to other fields, the importance of data efficiency and model interpretability grows increasingly important both for generalizability and for trusting AI systems.

Limitations. In our experiments with up-weighting the loss, we scale the loss for each low-frequency class by the same constant factor. Another option is to scale each class by a factor that is proportional to the relative frequency of the class. Then, among the low-frequency classes, the lowest-frequency class would be up-weighted more than the more-frequent classes. However, Byrd & Lipton still find that this does not improve performance (5).

Our model architectures used CLIP and ResNet-18 encoders pretrained on ImageNet (47). Satellite imagery comes from a starkly different domain than ImageNet, so one potential improvement would be to train the encoders. However, previous work has shown that pretraining is still beneficial, and we did use the ADE20K dataset which comes from a similar domain as ImageNet.

Acknowledgements. This project is conducted as part of another work and submitted for the COS 514 *Fundamentals of Deep Learning* course at Princeton University. The author(s) acknowledge that the work reported on in this paper was substantially performed using the Princeton Research Computing resources at Princeton University which is consortium of groups led by the Princeton Institute for Computational Science and Engineering (PICSciE) and Office of Information Technology's Research Computing. The authors would also like to acknowledge Professor Sanjeev Arora, TA Yikai Wu, and the COS 514 cohort for providing helpful feedback early in the project.

References

[1] Vighnesh Birodkar, Hossein Mobahi, and Samy Bengio. Semantic Redundancies in Image-Classification Datasets: The 10

[2] Laura Brenskelle, Rob P Guralnick, Michael Denslow, and Brian J Stucky. Maximizing human effort for analyzing scientific images: A case study using digitized herbarium sheets. *Applications in plant sciences*, 8(6):e11370–e11370, July 2020.

[3] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[4] Joy Buolamwini and Timnit Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91. PMLR, 23–24 Feb 2018.

[5] Jonathon Byrd and Zachary Chase Lipton. What is the Effect of Importance Weighting in Deep Learning? In *Proceedings of the 36th International Conference on Machine Learning*, pages 872–881. PMLR, 2019.

[6] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset Distillation by Matching Training Trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10718–10727, June 2022.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002. arXiv:1106.1813 [cs].

[8] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[9] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-Balanced Loss Based on Effective Number of Samples, Jan. 2019. arXiv:1901.05555 [cs].

[10] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. Cost-Sensitive Learning. In Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera, editors, *Learning from Imbalanced Data Sets*, pages 63–78. Springer International Publishing, Cham, 2018.

[11] R. Fong and A. Vedaldi. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In *International Conference on Computer Vision (ICCV)*, 2017.

[12] Benjamin Goldenberg, Burak Uzkent, Christian Clough, Dennis Funke, Deven Desai, grischa, Jesus Martinez Manso, Kat Scott, Meg Risdal, Mike Ryan, Pete, Rachel Holm, Ramesh Nair, Sean Herron, Tony Stafford, and Wendy Kan. Planet: Understanding the Amazon from Space, 2017.

[13] Joonho Gong and Hyunjoong Kim. RHSBoost: Improving classification performance in imbalance data. *Computational Statistics & Data Analysis*, 111:1–13, July 2017.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[15] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A Dataset for Large Vocabulary Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[17] Matthew Herland, Taghi M. Khoshgoftaar, and Richard A. Bauder. Big Data fraud detection using multiple medicare data sources. *Journal of Big Data*, 5(1):29, Sept. 2018.

[18] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The iNaturalist Species Classification and Detection Dataset. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8769–8778, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society.

[19] D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

[20] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[21] Tri Huynh, Aiden Nibali, and Zhen He. Semi-supervised learning for medical image classification using imbalanced training data. *Computer Methods and Programs in Biomedicine*, 216:106628, Apr. 2022.

[22] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A Survey on Contrastive Self-Supervised Learning. *Technologies*, 9(1), 2021.
[23] Chiyu Max Jiang, Mahyar Najibi, Charles R. Qi, Yin Zhou, and Dragomir Anguelov. Improving the intra-class long-tail in 3d detection via rare example mining. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 158–175, Cham, 2022. Springer Nature Switzerland.
[24] Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.
[25] H. Kahn and A. W. Marshall. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953.
[26] Vishal Kaushal, Rishabh Iyer, Suraj Kothawade, Rohan Mahadev, Khoshrav Doctor, and Ganesh Ramakrishnan. Learning From Less Data: A Unified Data Subset Selection and Active Learning Framework for Computer Vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1289–1299, 2019.
[27] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. Undoing the damage of dataset bias. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 158–171, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
[28] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
[29] M. Kubat. Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. *Fourteenth International Conference on Machine Learning*, June 2000.
[30] Miroslav Kubat, Robert C. Holte, and Stan Matwin. Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*, 30(2):195–215, Feb. 1998.
[31] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world, 2017.
[32] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.
[33] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection, Feb. 2018. arXiv:1708.02002 [cs].
[34] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.
[35] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-Scale Long-Tailed Recognition in an Open World. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
[36] Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, July 2019.
[37] Aravindh Mahendran and Andrea Vedaldi. Understanding Deep Image Representations by Inverting Them. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
[38] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
[39] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset Distillation with Infinitely Wide Convolutional Networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 5186–5198. Curran Associates, Inc., 2021.
[40] Sarah Parisot, Pedro M. Esperança, Steven McDonagh, Tamas J. Madarasz, Yongxin Yang, and Zhenguo Li. Long-Tail Recognition via Compositional Knowledge Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6939–6948, June 2022.
[41] Ronaldo C. Prati, Gustavo E. A. P. A. Batista, and Diego F. Silva. Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowledge and Information Systems*, 45(1):247–270, Oct. 2015.
[42] 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 Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.
[43] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data Distillation: Towards Omni-Supervised Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
[44] R. Bharat Rao, Sriram Krishnan, and Radu Stefan Niculescu. Data mining for improved cardiac care. *ACM SIGKDD Explorations Newsletter*, 8(1):3–10, June 2006.
[45] Nikita Rom. planets_dataset, 2019.
[46] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo Method*. Wiley Publishing, 3rd edition, 2016.
[47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. ImageNet Large

Scale Visual Recognition Challenge. 2015.
[48] Noveen Sachdeva and Julian McAuley. Data Distillation: A Survey. *Transactions on Machine Learning Research*, 2023. Survey Certification.
[49] Vignesh Sampath, Iñaki Maurtua, Juan José Aguilar Martín, and Aitor Gutierrez. A survey on generative adversarial networks for imbalance problems in computer vision tasks. *Journal of Big Data*, 8(1):27, 2021.
[50] Jéssica S. Santos, Rodrigo S. Ferreira, and Viviane T. Silva. Evaluating the classification of images from geoscience papers using small data. *Applied Computing and Geosciences*, 5:100018, 2020.
[51] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *International Conference on Computer Vision (ICCV)*, 2017.
[52] Vivswan Shitole, Fuxin Li, Minsuk Kahng, Prasad Tadepalli, and Alan Fern. One Explanation is Not Enough: Structured Attention Graphs for Image Classification. In *Neural Information Processing Systems (NeurIPS)*, 2021.
[53] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting, Sept. 2019. arXiv:1902.07379 [cs, stat].
[54] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations (ICLR) Workshops*, 2014.
[55] Nguyen Thai-Nghe, Andre Busche, and Lars Schmidt-Thieme. Improving Academic Performance Prediction by Dealing with Class Imbalance. In *2009 Ninth International Conference on Intelligent Systems Design and Applications*, pages 878–883, Nov. 2009. ISSN: 2164-7151.
[56] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. *A Deeper Look at Dataset Bias*, pages 37–55. Springer International Publishing, Cham, 2017.
[57] Angelina Wang and Olga Russakovsky. Overcoming Bias in Pretrained Models by Manipulating the Finetuning Dataset. 2023.
[58] Shanshan Wang, Cheng Li, Rongpin Wang, Zaiyi Liu, Meiyun Wang, Hongna Tan, Yaping Wu, Xinfeng Liu, Hui Sun, Rui Yang, Xin Liu, Jie Chen, Huihui Zhou, Ismail Ben Ayed, and Hairong Zheng. Annotation-efficient deep learning for automatic medical image segmentation. *Nature Communications*, 12(1):5915, 2021.
[59] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation, 2020.
[60] Wei Wei, Jinjiu Li, Longbing Cao, Yuming Ou, and Jiahang Chen. Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web*, 16(4):449–475, July 2013.
[61] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2023.
[62] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE International Conference on Data Mining*, pages 435–442, Nov. 2003.
[63] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4L: Self-Supervised Semi-Supervised Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
[64] Yaochi Zhao, Sen Chen, Qiong Chen, and Zhuhua Hu. Combining Loss Reweighting and Sample Resampling for Long-Tailed Instance Segmentation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, June 2023. ISSN: 2379-190X.
[65] Zhiyuan Zhao and Gauri Joshi. A Dynamic Reweighting Strategy For Fair Federated Learning. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8772–8776, May 2022. ISSN: 2379-190X.
[66] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018.
[67] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing through ADE20K Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, July 2017. ISSN: 1063-6919.
[68] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing Long-Tail Distributions of Object Subcategories. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, 2014.

Appendix

1. Loss up-weighting results

These are the loss up-weighting results on ADE20K for the remaining classes.

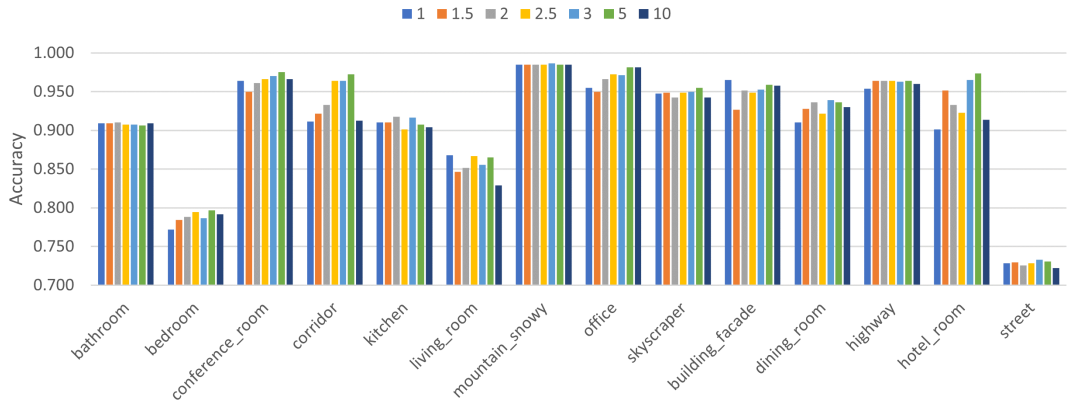
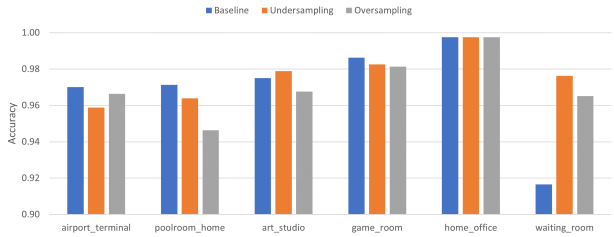


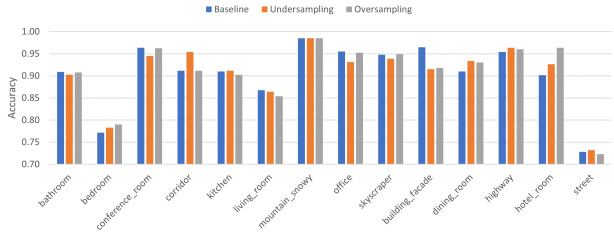
Figure 11: **Up-weighting accuracy on the remaining classes in ADE20K when using CLIP.**

2. Additional Experimentation

Previously, we tested a variant of undersampling and oversampling on ADE20K which is not described in the main paper. These results are not related to the methods or results presented in the main paper. Further, the results do not show a clear trend and do not contribute substantially towards our understanding; they are included for completeness.



(a) **Resampling accuracy on ADE20K's low-frequency classes with CLIP encoder.**



(b) **Resampling accuracy on ADE20K's remaining classes with CLIP encoder.**

Figure 12: **Resampling accuracy on the ADE20K dataset.**

Since ADE20K involves single-label classification, the resulting distributions were exactly uniform, using 98 for undersampling and 2, 241 for oversampling.