UM EECS 542: Advanced Topics in Computer Vision
UM CSE 598: Action and Perception
Homework #3: Classification with Imbalanced Pseudo-labels
Due: 8 October 2025 11:59pm

1 Overview

In this assignment, we will evaluate the imbalance in pseudo-labels generated by CLIP and consider a simple method, debiased learning with logit-adjustment, to address it.

Foundation models are an important tool for semi-supervised learning where they are used to generate pseudo-labels for weakly augmented data. Pseudo-labels with high confidence (above some threshold) may then be used to supervise training with strongly augmented data.

However, not only can foundation models be trained on imbalanced data, leading to biases in pseudo-labeling, but they can also produce imbalanced pseudo-labels even when their training data was balanced [1]. One strategy to address this is to debias pseudo-labels following the logit adjustment method [1].

2 Debiasing pseudo-labels

2.1 Logit adjustment

Real world data naturally follows a long-tailed distribution: the power law. When the ground-truth labels are imbalanced, traditional classification models perform worse on examples in the tail. However, we often believe that performance on rare classes is equally important to performance on common classes.

Multiple approaches exist to improve tail accuracy without significantly hurting performance on dominant classes: for example, resampling data, adjusting loss and learning rate, choosing specialized architectures, and post-training calibration [2]. One simple strategy is the logit adjustment proposed by [3].

Logit adjustment is based on the concept of a Bayes-optimal classifier which has the lowest average per-class error (every class is equally weighted in the average, regardless of its frequency in the real data). We would like to achieve such a classifier given constraints on the model we have. [4] showed that with a known prior probability distribution over labels, predicting label k for input x as

$$k = \underset{j}{\operatorname{argmax}} \frac{P(y = j \mid x)}{P(y = j)} = \underset{j}{\operatorname{argmax}}\, P^{\text{bal}}(y = j \mid x)$$

would minimize such average per-class error.

When a model is trained with cross-entropy loss, which is Bayes consistent, ideally, its output matches the conditional predicted class probabilities $P(y = j \mid x)$. So, in logit adjustment, we divide model outputs by the probability priors, transforming them into balanced conditional probabilities $P^{\text{bal}}(y \mid x)$ as defined above.

Assuming that model outputs are derived from logits via the softmax function (exponential relationship), at test time, we can balance the logits $s(x)$ as:

$$s^{\text{bal}}(x) = s(x) - \log P(y).$$

We can also learn $s^{\text{bal}}(x)$ directly during training by adding $\log P(y)$ to the logits before taking the loss. Finally, sometimes a scaling factor λ is multiplied into the adjustment as $\lambda \log P(y)$. However, this is theoretically unnecessary; empirically it usually provides little benefit.

Note: 'Bayes consistent' says that nearly optimal minimizers of cross entropy are nearly optimal minimizers of the average per-class error. However, models are often substantially imperfect (they represent a restricted set of functions so often converge to behave overconfidently) and the convergence patterns of minimizing cross entropy versus minimizing average per-class error differ [5].

2.2 Logit adjustment for pseudo-labels

We consider the FixMatch [6] self-supervised learning approach (details in Figure 1). We start from a foundation model such as CLIP and generate pseudo-labels for data missing ground truth assignments. However, we may know that our data is uniformly distributed, and we would like to use that knowledge to debias its pseudo-labels.

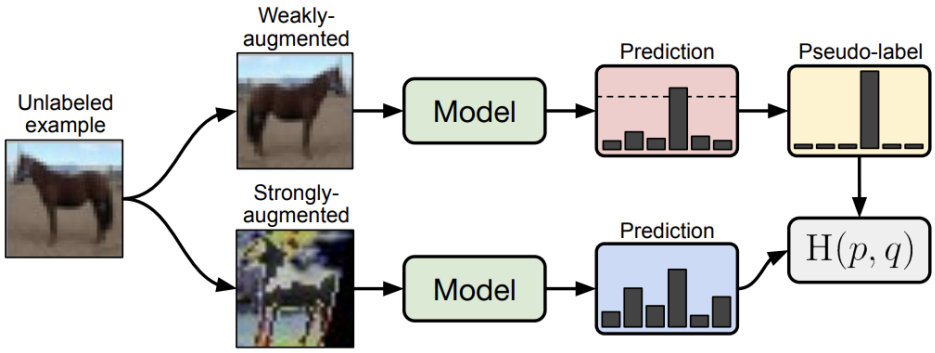


Figure 1: FixMatch approach for unlabeled data. Weakly-augmented images pass through the model to be trained. For each image, if its highest-predicted class is above a chosen threshold, a one-hot pseudo-label is generated associating that image with its predicted class. A cross entropy loss is then calculated between the pseudo label and the model prediction for a strongly augmented version of the same image. If the original prediction from the weakly-augmented image never surpassed the threshold, that image is not considered in the loss. If some labeled images are also provided, they contribute to the loss with standard cross entropy between ground-truth and their predicted labels.

One strategy is to determine the predicted pseudo-label distribution and use it to adjust the logits [1]. To make the approach less computationally expensive, and thus usable during self-supervised learning, we can estimate the pseudo-label probability distribution as a moving average.

Over a batch, update the pseudo-label probability distribution as:

$\hat{p} \leftarrow m\hat{p} + (1-m)$ (average of model output probabilities above threshold without debiasing).

The predicted logits $s(x)$ are then debiased as $s(x) - \log \hat{p}$.

2.3 Adaptive marginal loss

The model started by generating biased pseudo-labels, so we want to focus on improving its ability to distinguish the less-frequently predicted classes from the most-frequently predicted classes. I.e., to counteract pseudo-label bias, we could push apart different classes differently.

In other words, the highest confusion is between classes the model is biased towards and classes the model is biased away from. So, if our loss emphasizes the separation between such classes, we would learn to distinguish them better, and supposedly reduce overall bias. [1] proposes an adaptive marginal loss for this (*optional*: see paper for precise formulation).

3 Problem Set

Our goal is to visualize the imbalance in predicted labels from CLIP and consider how it affects per-class precision, recall, and inter-class confusion. We will then consider how self-supervised learning with debiasing as in [1] changed performance. There is no model training to do.

1. (60 points) Following the starter code, predict labels for the ImageNet100 validation set using the standard pretrained ResNet50 CLIP.

   (a) (20 points) Follow the format of Figure 2, but use the ImageNet100 validation set. Create a histogram with the number of predictions per class on the y-axis and the classes arranged from

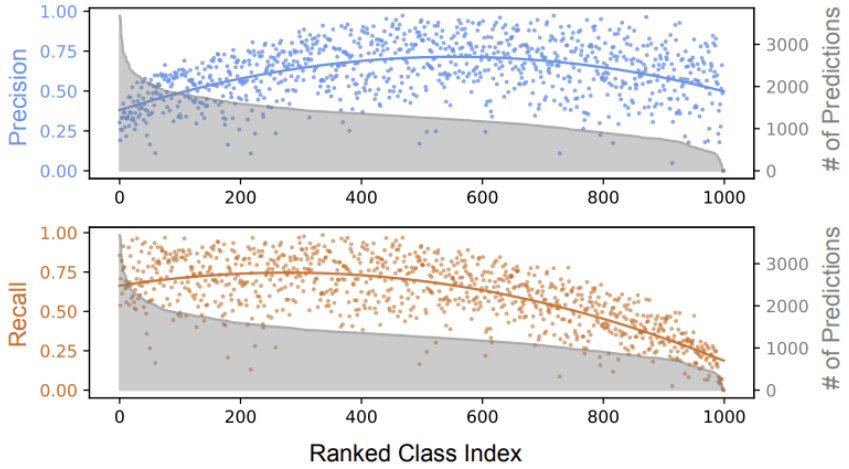


Figure 2: Example of graph to be made. This graph shows the pseudo-label predictions of CLIP for ImageNet1K. You need to replicate it for ImageNet100.

most to least predicted on the x-axis. Duplicate this histogram and overlay on the first copy the precision and on the second copy the recall for each class.

(b) (20 points) Follow the format of Figure 3, but use the ImageNet100 validation set. Create a confusion matrix for all 100 classes. Order the classes from top to bottom based on the total number of images predicted to be in that class. I.e., the first row is the most predicted class and the last row is the least predicted class.

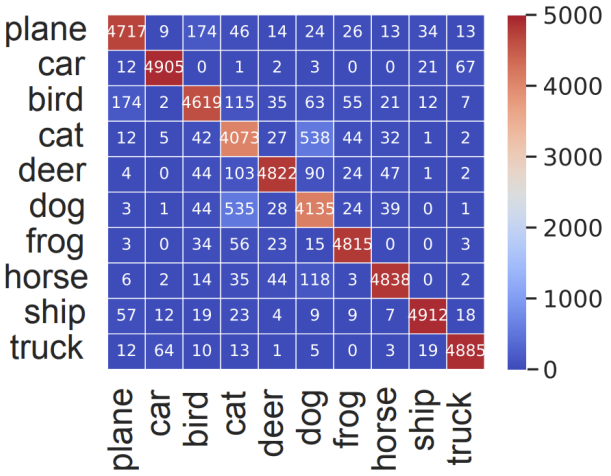
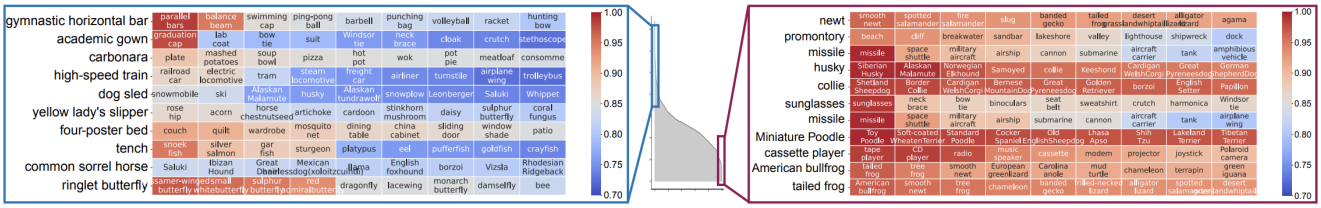


Figure 3: Example of confusion matrix to be made. This matrix is for the pseudo-label predictions of CLIP on CIFAR10. You need to replicate it for ImageNet100.

(c) (20 points) Get the centroid of the logits of every class. Following the model of Figure 4, compare the single most predicted and the single least predicted classes: for each of the two classes, get the closest 10 classes (by cosine similarity of centroids) and output the cosine similarity in an ordered vector from highest to lowest value. Submit the **names of the most and least predicted classes, the 2 vectors, and the ordered list of classes corresponding to each**, no need to make a figure.



a) top-10 classes b) least-10 classes

Figure 4: Comparison of logit similarity between (a) most and (b) least frequently predicted ImageNet1K classes with CLIP. The closest 10 classes (by average logit cosine similarity) are arranged from highest to lowest similarity. The less frequently predicted classes usually have strong inter-class correlations, while the most frequently predicted classes are the opposite.

2. (60 points) Following the starter code, use ImageNet100 to evaluate the results of ResNet50 CLIP that was fine-tuned with the debiasing self-supervised approach [1] and no labeled data (zero-shot) on ImageNet1K.

(a) (20 points) Create the same histograms as in question 1: follow the format of Figure 2. **Keep the same order of classes as in Q1** even if, after logit adjustment, the relative number of

predictions between classes has shifted.

(b) (20 points) Create the same matrix as in question 1: follow the format of Figure 3. Create a confusion matrix for all 100 classes. **Keep the same order of classes as in Q1** even if, after logit adjustment, the relative number of predictions between classes has shifted.

(c) (20 points) Follow the same process as in question 1. Get the centroid of the logits of every class. Following the idea in Figure 4, compare the **single most predicted, the single least predicted, and also the same two classes you identified in Q1**. For each of the **FOUR** classes, get the closest 10 classes (by cosine similarity of centroids) and output the cosine similarity in an ordered vector from highest to lowest value. Submit the names of the considered classes, the four vectors, and the ordered list of classes corresponding to each.

4 Setup

1. Clone and follow the installation steps in the repository: `https://github.com/frank-xwang/debiased-pseudo-labeling`
2. In the repository root, place the provided `starter.py`. You can implement everything in this file.
3. Download the reproduced Zero-Shot DebiasPL checkpoint from `debiaspl_zsl_epoch50.pth.tar` linked from the repo.
4. `ImageNet100` dataset is provided. You should be able to see it at `/scratch/eecs542f25_class_root/eecs542f25_class/shared_data/imagenet-100`. (`cse598f25s014_class` if you are in CSE598) **Please do not modify anything in the dataset folder.**

5 Submission Instructions

1. This assignment is to be completed individually.
2. Submissions should be made through Gradescope and Canvas. Please make sure you submit to the right course session you are enrolled in. **Please upload the PDF to Gradescope and the ZIP file to Canvas**:

   a) A **PDF file** for the **report**: At the top of the first page, include your name, student ID, and the date of submission. The report should contain all graphs and the requested vectors/lists, clearly labeled by sub-question.

   If you believe there may be an error in your code, provide a written statement in the PDF explaining what might be wrong and how it affected your results. For any functionality/module you were unable to implement, you may include pseudocode and/or expected results. Any additional explanations are optional and, if included, should be presented as a fifth page at the end of the report.

   b) A **ZIP file** for the **code implementation**: Submit a ZIP file containing a folder with all your code (exclude model weights). You may refactor or add files as needed. If you make substantial changes, include comments and reasonable filenames. It's acceptable to display figures interactively (e.g., `imshow`) rather than saving on first run. Grading includes a quick code review for reasonable implementations.

References

[1] Xudong Wang, Prateek Mitra, Stella Yu, Alexei A. Efros, and Trevor Darrell. Debiased learning from naturally imbalanced pseudo-labels. In *CVPR*, pages 14647–14657, 2022.

[2] Chongsheng Zhang, Qitian Yin, and Zhi-Hua Zhou. A systematic review on long-tailed learning, 2024.

[3] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment, 2021.

[4] Guillem Collell, Drazen Prelec, and Kaustubh Patil. Reviving threshold-moving: a simple plug-in bagging ensemble for binary and multiclass imbalanced data, 2017.

[5] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications, 2023.

[6] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin D. Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence, 2020.