

# Text-Guided Mixup Towards Long-Tailed Image Categorization-Supplementary

Richard Franklin<sup>1</sup>  
rsamfranklin@gmail.com

Jiawei Yao<sup>1</sup>  
jwyao@uw.edu

Deyang Zhong<sup>1</sup>  
dyzhong@uw.edu

Qi Qian<sup>2</sup>  
qi.qian@alibaba-inc.com

Juhua Hu<sup>1</sup>  
juhuah@uw.edu

<sup>1</sup> School of Engineering and Technology  
University of Washington  
Tacoma, WA

<sup>2</sup> Alibaba Group  
Bellevue, WA

## 1 Effect of Local Sampling

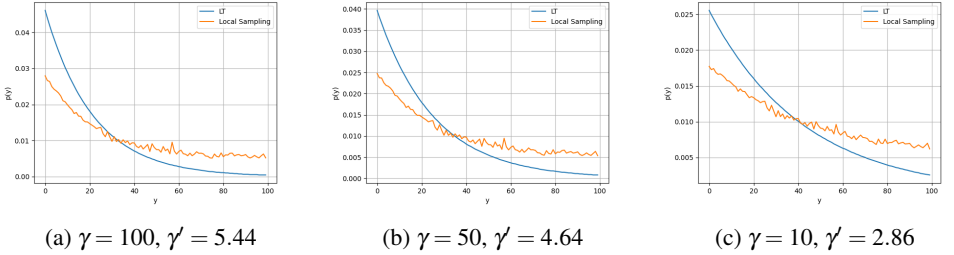


Figure 1: Local sampling effect on CIFAR100-LT  $p(y)$  distribution

As discussed in the main paper, at each training step, local sampling feeds the model an image pair that holds semantically-related images, where the semantic relation is determined by the text encoder. In constructing the pair, the label of the first image is determined by

$$p(y = y_i) = \frac{n_i}{\sum_{k=1}^C n_k} \quad (1)$$

which is to uniformly sample an image without replacement from the dataset. However, the label of the second image is determined by

$$p_{ls}(y = y_j | y_i) = \begin{cases} \frac{\exp(f_{T_i} \cdot f_{T_j} / \tau)}{\sum_{k=1}^C \exp(f_{T_i} \cdot f_{T_k} / \tau)} & i \neq j \\ 0 & \text{o.w.} \end{cases} \quad (2)$$

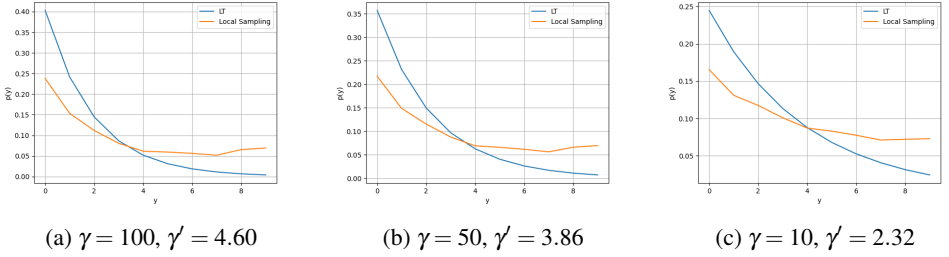


Figure 2: Local sampling effect on CIFAR10-LT [14]  $p(y)$  distribution

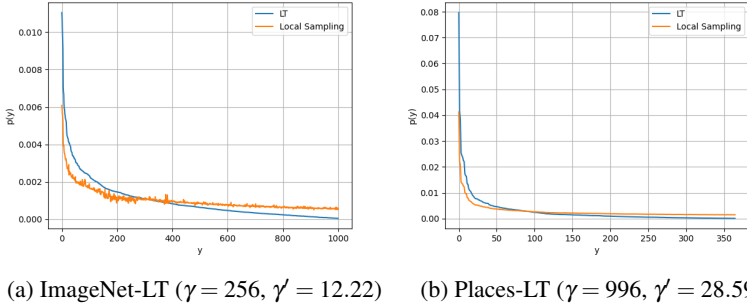


Figure 3: Local sampling effect on ImageNet-LT [14] and Places-LT [9]  $p(y)$  distribution

which ignores the sample count for any class label. Due to the negligence of the second label’s sample count, the amount of times that the model sees minority classes can be increased effectively balancing the data distribution by resampling. To observe the amount of resampling, we show the sample count before and after local sampling as follows. Allow  $Y$  to be the random variable in the event that local sampling yields an instance of class  $y \in \{y_i, y_j\}$ , and allow  $y_i$  to be the event that  $y_i = y$  and  $y_j$  to be the event that  $y_j = y$ . The probability that the model observes an image with class label  $y$  can be calculated as

$$\begin{aligned}
 p(Y = y) &= p(y_i) + (1 - p(y_i))p(y_j) \\
 &= p(y_i) + (1 - p(y_i)) \sum_{k, k \neq i}^C p(y_j | y_k) p(y_k).
 \end{aligned}$$

Using Eqns. 1 and 2,  $p(Y)$  can be evaluated for all  $y$ , and we illustrate the resulting  $p(y)$  for every dataset in Figs. 1-3. Additionally, we indicate the new imbalance factor as  $\gamma'$ . We can observe that the imbalance severity and the magnitude of long-tailed distribution can be well reduced, which demonstrates the effectiveness of our local sampling method.

## 2 Comparison between Textual Similarity and Visual Categorization

To further confirm our assumption that semantically related classes are visually related, we make a comparison between class label textual similarities and CLIP’s zero-shot performance.

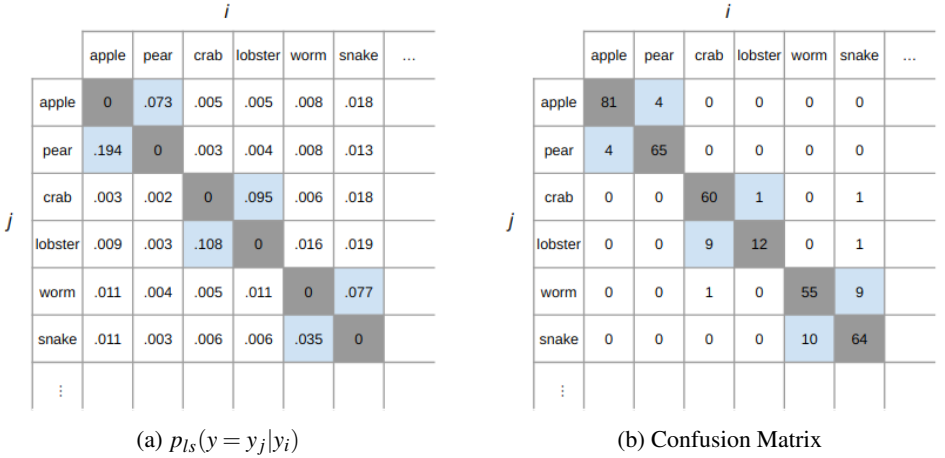


Figure 4: The tables above demonstrate the correlation between text feature similarities (captured by  $p_{ls}$ ) and the model performance with zero-shot classification. The diagram on the left shows our constructed probability distribution  $p_{ls}$  for CIFAR100 [14], and the diagram on the right is a confusion matrix of CLIP’s performance on CIFAR100 without training. The columns represent class  $y_i$ , and the rows represent class  $y_j$ . For demonstration purposes, we present three pairs of related classes: (apple, pear), (crab, lobster), and (snake, worm). Blue cells hold values for related class pairs while gray cells can be ignored since they hold the values for same class pairs. It can be observed that the blue cells hold values that are generally higher than any of the other white cells in their respective rows.

Fig. 4 shows a comparison between our semantic probability distribution  $p_{ls}$  and a confusion matrix of CLIP’s zero-shot classification performance using CIFAR100’s validation set. It can be observed that  $p_{ls}$  is correlated with the performance of zero-shot classification. By observing the blue cells in the confusion matrix, we see that the model more frequently struggles to find a decision boundary between related classes. When we sample with  $p_{ls}$ , we expect that we are sharing information with related classes more frequently and thus establish a decision boundary more optimally positioned for inference on the balanced validation data.



Figure 5: An illustration of the theorized effect that label shift has on the model’s decision boundary. Red circles indicate feature vectors of tail classes and green circles are that of nearby head classes. When  $\alpha > 0$ , the decision boundary shifts towards the head classes anticipating for higher intra-class variance for tail classes.

### 3 Algorithms and Training Configurations

In this section, we summarize the algorithms for LocalSample, Mix, and the entire training process, where the effect on the proposed mixup technique on the decision boundary between nearby head and tail classes is illustrated in Fig. 5. Upon acceptance of this paper, we will also publicly release the code.

---

**Algorithm 1** LocalSample ( $\tau, f_T, D = \{(x, y)\}$ )
 

---

```

1:  $p_{y_i} \leftarrow [0, 1]^C$  vector representing the probability distribution from Eqn. 1
2:  $p_{y_j|y_i} \leftarrow [0, 1]^{C \times C}$  matrix representing the probability distribution from Eqn. 2 with given  $\tau$  and  $f_T$ 
3: while model is not converged do
4:    $y_i \sim p_{y_i}$ 
5:    $y_j \sim p_{y_j|y_i}$ 
6:    $x_i \sim \{x \mid (x, y) \in D \text{ and } y = y_i\}$ 
7:    $x_j \sim \{x \mid (x, y) \in D \text{ and } y = y_j\}$ 
8:   yield  $(x_i, y_i), (x_j, y_j)$ 
9: end while
  
```

---



---

**Algorithm 2** Mix ( $\alpha, (x_i, y_i), (x_j, y_j)$ )
 

---

```

1: Convert  $y_i$  and  $y_j$  to one-hot vectors of size  $C$ 
2:  $\lambda_x \sim \text{Beta}(0.5, 0.5)$ 
3:  $\lambda_y \leftarrow$  label shift assignment by Eqn. 2 in the main paper
4:  $x^{LFM} \leftarrow \lambda_x x_i + (1 - \lambda_x) x_j$ 
5:  $y^{LFM} \leftarrow \lambda_y y_i + (1 - \lambda_y) y_j$ 
6: return  $x^{LFM}, y^{LFM}$ 
  
```

---

During the training, we use the hyperparameters and other training properties listed in Table 1. Most experiments have the same setup, but some minor adjustments are made largely due to differences in class label distributions. Under the circumstances of heavy class imbalance, we can simply raise the values of  $\alpha$  and  $\tau$ , which we do for Places-LT [9]. Detailed information for each dataset is provided in Table 2. The original dataset imbalance is summarized by the imbalance factor  $\gamma$ .

### 4 Additional Ablation Studies

Besides the ablation study conducted in the main paper, we also conducted the following ablation studies.

#### 4.1 Effect of $\alpha$

We study the effect of the intensity in which we shift the training label assigned to each mixup, for which we can control with  $\alpha$ . The  $\alpha$  value directly affects the positioning of the model’s decision boundaries between class pairs, and we can expect lower values to extend the boundary of many-shot classes and higher values to extend the boundary of few-shot classes.

**Algorithm 3** Train ( $\mathcal{F}_T, \mathcal{F}_I, W_I, \alpha, \tau, D, T$ )

---

```

1: Initialize  $\Theta_T$  and  $\Theta_I$  (weights of  $\mathcal{F}_T$  and  $\mathcal{F}_I$ , respectively) with pre-trained weights
2: Freeze  $\Theta_T$ 
3:  $f_T \leftarrow \Pi_{\|\cdot\|_2=1} \mathcal{F}_T(T)$ 
4: for epoch in  $1, \dots, N_0$  do ▷ Stage 1
5:   for  $(x_i, y_i), (x_j, y_j)$  in LocalSample ( $\tau, f_T, D$ ) do
6:      $x^{LFM}, y^{LFM} \leftarrow \text{Mix}(\alpha, (x_i, y_i), (x_j, y_j))$ 
7:      $f_I \leftarrow \Pi_{\|\cdot\|_2=1} \mathcal{F}_I(x^{LFM})$ 
8:      $\ell \leftarrow \mathcal{L}(f_T \cdot f_I, y^{LFM})$ 
9:     Update  $\Theta_I$ 
10:  end for
11: end for
12: Freeze  $\Theta_I$  ▷ Stage 2
13: Initialize  $W_I$  as  $d \times d$  identity matrix,  $I_d$ , where  $d$  is the feature dimension of  $\mathcal{F}_I$ 
14: for epoch in  $1, \dots, N_1$  do
15:   for  $(x_i, y_i), (x_j, y_j)$  in LocalSample ( $\tau, f_T, D$ ) do
16:      $x^{LFM}, y^{LFM} \leftarrow \text{Mix}(\alpha, (x_i, y_i), (x_j, y_j))$ 
17:      $f_I \leftarrow \Pi_{\|\cdot\|_2=1} (W_I^T \mathcal{F}_I(x^{LFM}))$ 
18:      $\ell \leftarrow \mathcal{L}(f_T \cdot f_I, y^{LFM})$ 
19:     Update  $W_I$ 
20:   end for
21: end for

```

---

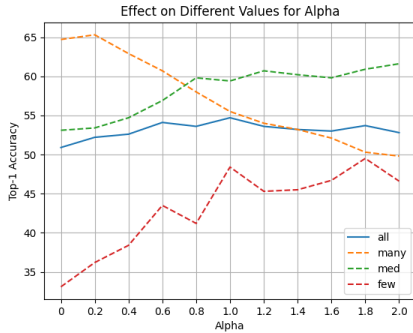
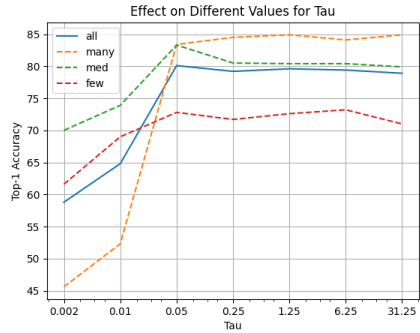
Table 1: Hyperparameters and configurations.

Dataset	CIFAR10-LT [■]		CIFAR100-LT [■]		ImageNet-LT [■]		Places-LT [■]	
Stage	1	2	1	2	1	2	1	2
Epochs	10	10	50	10	30	10	30	10
Learning Rate	$1 \times 10^{-9}$	$5 \times 10^{-1}$	$1 \times 10^{-6}$	$1 \times 10^{-2}$	$5 \times 10^{-6}$	$1 \times 10^{-2}$	$1 \times 10^{-7}$	$5 \times 10^{-4}$
LR Scheduler	Cosine Annealing		Cosine Annealing		Cosine Annealing		Cosine Annealing	
Min LR	$1 \times 10^{-12}$	$5 \times 10^{-4}$	$1 \times 10^{-9}$	$1 \times 10^{-5}$	$5 \times 10^{-9}$	$1 \times 10^{-5}$	$1 \times 10^{-10}$	$5 \times 10^{-7}$
Optimizer	Adam		Adam		Adam		Adam	
Batch Size	32		32		96		96	
$\alpha$ for LFM	1.00		1.00		1.00		1.25	1.50
$\tau$ for LFM	0.05		0.05		0.05		1.00	
Seed	0		0		0		0	

Table 2: Detailed information of mentioned datasets

Dataset	CIFAR10-LT [■]			CIFAR100-LT [■]			ImageNet-LT [■]	Places-LT [■]
Number of classes	10			100			1000	365
Total Training Images	20,431	13,996	12,406	19,573	12,608	10,847	115,846	62,500
Max Images	5,000	5,000	5,000	500	500	500	1,280	4,980
Min Images	500	100	50	50	10	5	5	5
Original Imbalance Factor $\gamma$	10	50	100	10	50	100	256	996
Effective Imbalance Factor $\gamma'$	2.32	3.86	4.60	2.86	4.64	5.44	12.22	28.59

In this study, we change  $\alpha$  among the range  $[0, 2]$  on CIFAR100-LT with an imbalance factor of 100 using CLIP’s ResNet50 backbone with the same configuration settings. From Fig. 6,

Figure 6: Effect of different  $\alpha$ .Figure 7: Effect of different  $\tau$ .

we can easily observe that an increasing of  $\alpha$  can slowly degenerate the performance of many-shot classes while improve the performance of the other, especially that of the few-shot classes as expected. The result also reveals that setting  $\alpha$  to 1 works best for all accuracies.

## 4.2 Effect of $\tau$

To study the effect of different temperature settings for  $p_{ls}$ , we run multiple experiments with  $\tau = \{.002, .01, .05, .25, 1.25, 31.25\}$ . At lower values, we increase the probability that nearby class samples ( $i, j$ ) are paired together. At higher values, the probability of two nearby class samples becoming paired is mitigated, and the class sampling becomes more balanced. We run our experiments on CIFAR100-LT [14] with an imbalanced factor of 100 using CLIP’s ViT-B/16 backbone, which is of the same configuration settings. Fig. 7 reveals that when we increase  $\tau$  from a small value, all classes can benefit from LFM by mixing semantically related samples, while after  $\tau = 0.05$  it plateaued. Therefore,  $\tau = 0.05$  is adopted in the rest of our experiments.

## References

- [1] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.
- [2] 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.
- [3] 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*, 2017.