Appendix for CurrMix

Zhongquan Jian[1,†], Yanhao Chen[2,†], Bingbing Hu[2], Wenhan Lv[2], Shaopan Wang[3], Jipeng Wu[1], Junfeng Yao[1,2,3,*], Yang Lu[3,*], Qingqiang Wu[2,3,*]
[1]School of Computer and Data Science, Minjiang University, Fuzhou, China
[2]School of Film, Xiamen University, Xiamen, China
[3]School of Informatics, Xiamen University, Xiamen, China
jianzq@mju.edu.cn, wuqq@xmu.edu.cn

A. Implementation Details

In our experiments, SGD with momentum 0.9 is employed as the optimizer, the weight decay rate is 5e-4, and the cosine annealing scheduler is utilized to gradually decrease the learning rate. For fair comparisons, ResNet-32 [2] is selected as the backbone for CIFAR-100/10-LT, ResNeXt-50 [5] for ImageNet-LT, and ResNet-50 [2] for iNaturalist2018, in line with most of the baselines.

For the CIFAR-10-LT and CIFAR-100-LT datasets, a data transform strategy is applied, involving random cropping of a 32×32 region from images padded with 4 pixels on each side and flipping with a 0.5 probability. Training runs for 200 epochs with a mini-batch size of 64 and an initial learning rate (lr) of 0.01. The hyperparameter β is set 2.0. *In the Alternating Curriculum Sampling (ACS) strategy, t_{curl} and t_{jump} are set to 10 and 185, respectively, while k', critical for Data Augmentation (DA), is set to 3 by default. For the adaptively adjust parameters, the initial number of images in each mix k_0 is set to 3, and the initial class weighting coefficient α_0 is set to 0.5 for CIFAR-100-LT and 0.7 for CIFAR-10-LT.*

For ImageNet-LT and iNaturalist 2018, we adopt the data transformation strategy from Li et al. [4], scaling the shorter dimension to 256 and randomly cropping a 224×224 patch from the augmented image or its horizontal flip. The training lasts 135 and 120 epochs, respectively, with mini-batch sizes of 128 and 512. The learning rate is initialized to 0.1 for both datasets. β is also set to 2.0 for both datasets. *Hyper-parameters t_{curl} and t_{jump} are set to 10 and 110. For ImageNet-LT, k_0 and α_0 are set to 3 and 0.4, while for iNaturalist2018, these values are 3 and 0.1. The hyper-parameter k' is fixed at 2 by default.*

The basic parameter settings used for different datasets are summarized in Table 1. **We validate the stability of the newly introduced ACS parameters by varying them within a reasonable range**. These results demonstrate that the proposed method is robust to hyperparameter settings, enabling its application to diverse scenarios in reality.

* Corresponding Authors, † Equal Contribution.

B. Additional Ablation Studies

B.1. Analysis of Progressive Training

B.1.1. Minority Class Adaption

As depicted in Eq. (8) in the main paper, α_t represents the class weighting coefficient, indicating the importance of minority classes, with a larger α_t assigns greater importance to them. Table 2 summarizes the experimental results for different methods across various α_t values, with Fig. 1 providing a visual illustration for clarity. The solid line in Fig. 1 represents our proposed method that gradually increases the class weighting coefficient α_t during training, *i.e.,* from α_0 to $\alpha_0 + 1.0$. In contrast, the dashed lines represent static methods that maintain fixed class weighting coefficients at the maximum, average, or minimum values, *i.e.,* $\alpha_0 + 1.0$, $\frac{\alpha_0+1.0}{2}$ and α_0.

In our ablation experiments, we vary α_0 from 0.1 to 0.6. As shown by the yellow line, compared with the green and red lines, consistently using larger class weighting coefficients, namely giving higher importance to minority classes, adversely affects the model's performance. Additionally, comparing the green and red lines with the blue line reveals that small class weighting coefficients fail to adequately emphasize the importance of minority classes, resulting inferior performance to that of our proposed progressive method.

B.1.2. Model Focus Transition

Fig. 2 illustrates the performance under different loss weighting coefficients β. β is a hyper-parameter that controls the transition speed from representation learning to classifier fine-tuning. $0 < \beta < 1$ means the classifier fine-tuning dominates, while $\beta > 1$ means the visual representation learning dominates. $\beta = 1$ means that the classifier fine-tuning and the visual representation learning are bal-

Table 1. Summary of parameter setting on different datasets.

Type	Parameter	Values			
		CIFAR-100-LT	CIFAR-10-LT	ImageNet-LT	iNaturalist 2018
GLMC [1]	Backbone	ResNet-32	ResNet-32	ResNeXt-50	ResNet-50
	Batch size	64	64	128	512
	Initial lr	1e-2	1e-2	1e-1	1e-1
	Weight decay	5e-4	5e-4	5e-4	5e-4
	Momentum	0.9	0.9	0.9	0.9
	Epochs	200	200	135	120
	β	2.0	2.0	2.0	2.0
ACS	α_0	0.5	0.7	0.4	0.1
	k_0	3	3	3	3
	t_{curl}	10	10	10	10
	t_{jump}	185	185	110	110
DA	k'	3	3	2	2

Table 2. Results under different initial class weighting coefficients. The second column represents results with our proposed progressive method ($\alpha_0 \rightarrow \alpha_0 + 1.0$), while the other three columns represent results with static methods that keep the class weighting coefficients at maximum ($\alpha_0 + 1.0$), average ($\frac{\alpha_0+1.0}{2}$), and minimum (α_0) values, respectively.

α_0	Progressive Increase	Keep at Maximum	Keep at Average	Keep at Minimum
0.1	54.67	56.08	50.14	52.05
0.2	55.03	56.38	50.36	52.35
0.3	55.85	56.18	50.88	52.57
0.4	57.28	52.31	51.90	52.85
0.5	57.95	45.00	52.02	53.07
0.6	54.23	38.58	52.35	53.44

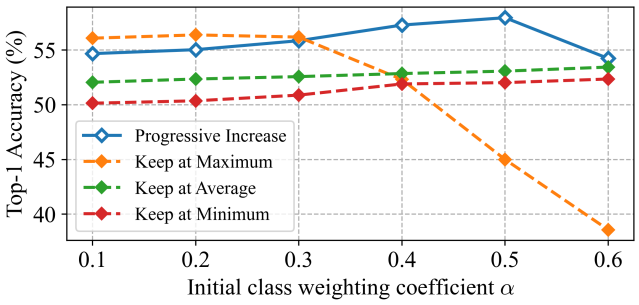

Figure 1. Ablations with different class weighting strategies.

anced, with linear transition from the former to the latter.

As shown in Fig. 2, the model achieves optimal performance at $\beta = 2$, highlighting the importance of balancing representation learning and classifier fine-tuning. Specifically, allocating more focus on representation learning in the early stages proves beneficial, aligning with the principles of two-stage learning [3, 6]. Nonetheless, proper classifier fine-tuning is equally essential. As shown in the figure, a significant increase in β drastically reduces the time allocated for classifier fine-tuning, leading to a sharp decline in model performance. Therefore, we also set $\beta = 2.0$ in our experiments.

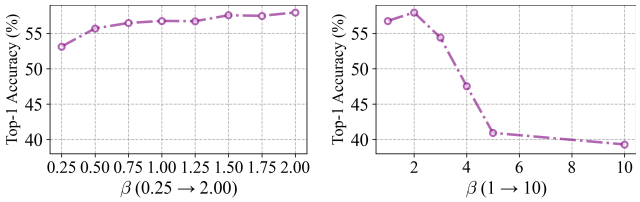

Figure 2. Ablations with different loss coefficients.

B.2. Analysis of the ACS Strategy

For the ACS strategy, two sampling strategies are defined: Easier Image-prioritized Sampling (EipS) and Harder Image-prioritized Sampling (HipS). In practice, the ACS strategy is implemented by alternating between EipS and HipS, *i.e.,* EipS $\leftrightarrow$ HipS and HipS $\leftrightarrow$ EipS. The training processes for different strategies are illustrated in Fig. 3. Note that, training accuracies are evaluated every t_{curl} epochs, leading to the step-shaped pattern in Fig. 3(c).

As shown in Fig. 3(a), EipS converges faster initially but slows later, while HipS follows the opposite trend. This leads to higher validation loss for EipS and lower validation loss for HipS (Fig. 3(b)), causing underfitting and overfitting, respectively (Fig. 3(c)). As shown by the red lines in Fig. 3(a) and Fig. 3(b), the alternating sampling strategies balance the convergence speed, with the curves posi-

Table 3. Results under different k', meaning that each image in the training set is generated k' times by mixing different images for model training.

k'	Accuracy(%)	Training
1	40.14	1.00×
2	56.46	1.55×
3	57.95	2.12×
4	56.89	2.70×
5	55.47	3.30×

tioned between those of EipS and HipS, leading to lower training and validation losses in the end of model training. Additionally, as shown in Fig. 3(c), the alternating sampling strategies avoid underfitting and effectively prevent overfitting. Finally, Fig. 3(d) highlights the clear advantages of alternating sampling strategies over independent sampling, achieving higher validation accuracy.

Moreover, influenced by the hyperparameter k_t, a step-jumping phenomenon appears in the training and accuracy curves at the t_{jump} epoch, with EipS exhibiting the largest jump and HipS the smallest. However, the smaller jump amplitude limits HipS's performance, while the larger amplitude fails to compensate for the underfitting issues in EipS during training. Therefore, the alternating sampling strategies effectively alleviate these issues and achieve better model performance, particularly HipS $\leftrightarrow$ EipS, which outperforms the other strategies.

B.3. Analysis of Data Augmentation

The hyper-parameter k' is crucial for CurMix's performance. While increasing k' boosts model accuracy by expanding the training data, it also significantly increases training time and affects loss backpropagation. Table 3 quantitatively evaluates the impact of varying k' on both performance and time cost. As observed, increasing k' from 1 to 2 results in a substantial improvement in model performance ($40.14\% \rightarrow 56.46\%$), though training time increases by 0.55 times. As k' continues to rise, the performance peaks at $k' = 3$ before declining. The potential reason may be that augmenting the samples enhances the model's ability to recognize mixed classes in the images, thereby affecting its original classification performance.

C. Visualization and Analysis

C.1. Analysis of Mixed Images

To alleviate the issue of data imbalance, MixUp serves as an effective data augmentation technique by blending multiple images to generate mixed samples. As illustrated in Fig. 4, when $k = 1$, no additional images are mixed, corresponding to the original, highly imbalanced class distribution. As k increases, the effective class distribution becomes progressively more balanced, thereby mitigating the imbalance issue. Hence, increasing k is theoretically beneficial. Ideally, when all classes are included within each mixed image, a perfectly balanced data distribution can be theoretically achieved, providing fairer training opportunities for all classes.

C.2. Analysis of Sampled Images

In ACS, image difficulty is assessed based on corrected classification probabilities, with higher probabilities indicating easier samples. These difficulty scores are recorded throughout model training without incurring additional computational overhead. Therefore, this raises an important question: are tail-class images inherently more difficult to classify and thus more likely to be sampled under the ACS strategy?

To explore this, we visualize the class distribution of sampled images in Fig. 5, where the x-axis represents the training process (epochs), the y-axis represents classes from the most head (0) to the most tail (99), and the color intensity indicates class frequencies of sampled images in each epoch. The visualization shows that easier images predominantly originate from head classes, especially in the later stages of training. For instance, during epochs 180–190, when the EipS sampling strategy is applied, a higher proportion of head-class images are selected as easier samples, as indicated by the darker regions in the heatmap. In contrast, during epochs 190–200, under the HipS strategy, the sampled harder images are more evenly distributed across all classes, with only a slight increase in head classes. Therefore, the harder samples are not consistently associated with tail classes across the training process. This suggests that ACS does not introduce a systematic sampling bias toward tail classes, but rather selects images based on difficulty regardless of class frequency.

References

[1] Fei Du, Peng Yang, Qi Jia, Fengtao Nan, Xiaoting Chen, and Yun Yang. Global and local mixture consistency cumulative learning for long-tailed visual recognitions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15814–15823, 2023. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[3] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations*, 2020. 2

[4] Mengke Li, Yiu-Ming Cheung, and Zhikai Hu. Key point sensitive loss for long-tailed visual recognition. *IEEE Trans-*

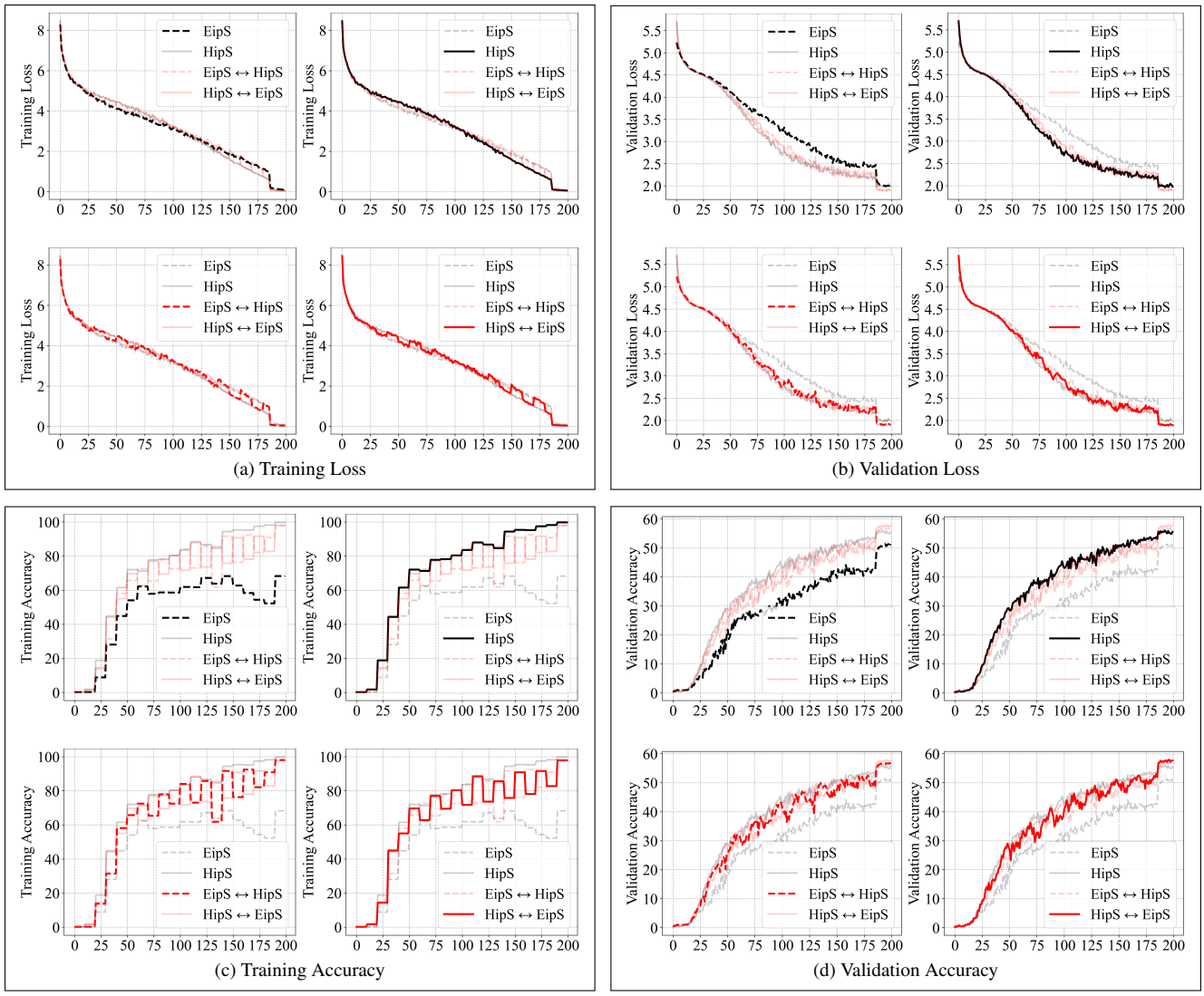

Figure 3. Illustrations of the training process under different strategies.

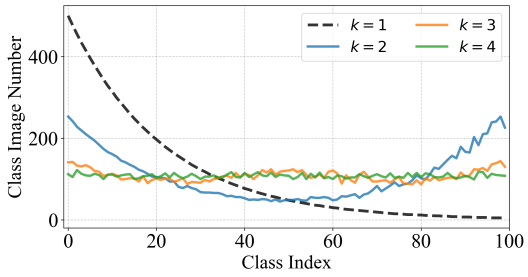

Figure 4. Mixing diverse images helps balance the class distribution, $k = 1$ means no additional images are mixed, *i.e.*, original class distribution. Ideally, a perfect class balance can be achieved when all classes are mixed in the image.

actions on Pattern Analysis and Machine Intelligence, 45(4): 4812–4825, 2023. 1

[5] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5987–5995, 2017. 1

[6] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2361–2370, 2021. 2

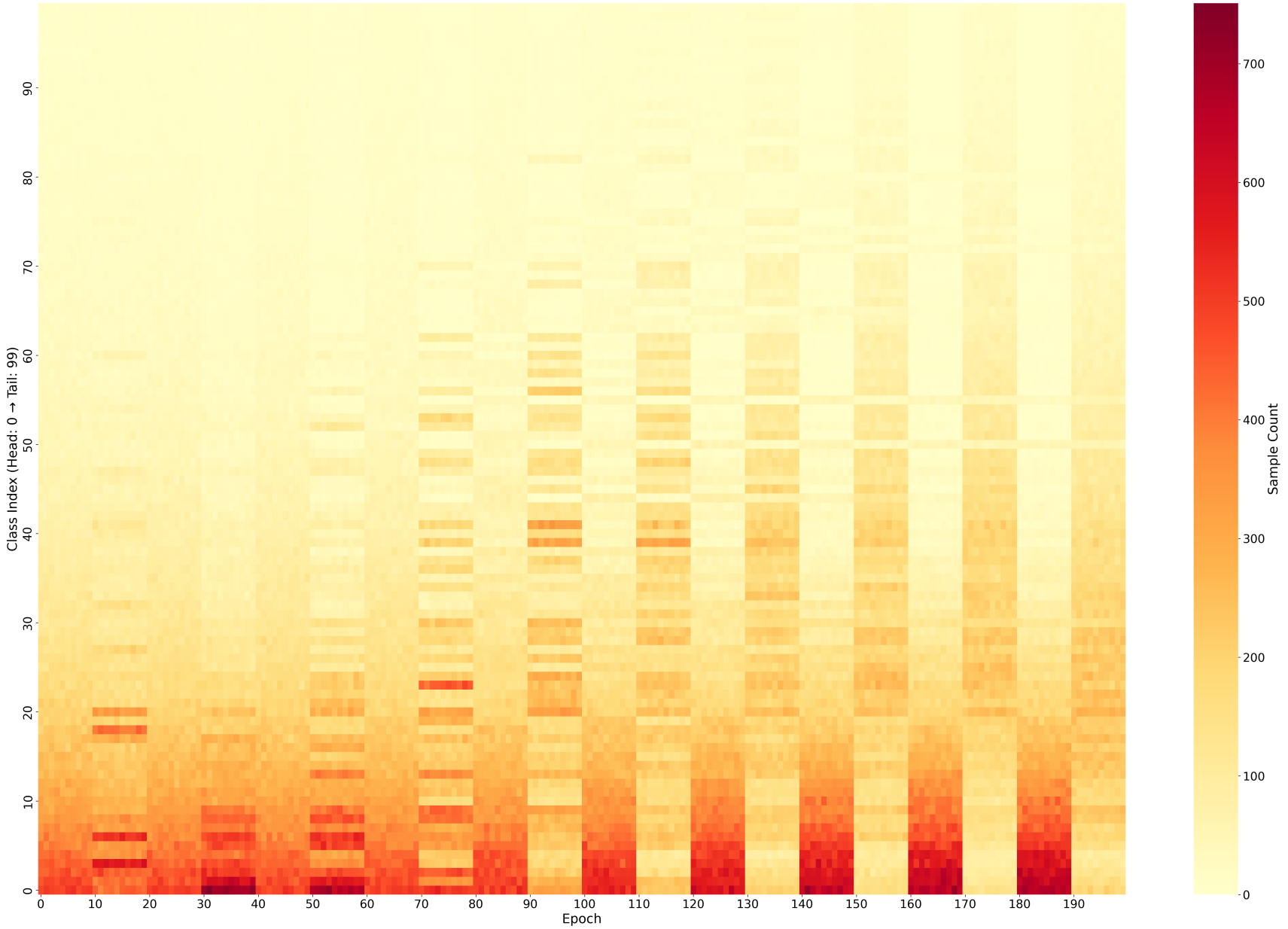

Figure 5. Illustration of the class distribution of sampled images during the training process. The x-axis represents the training process (epochs), the y-axis represents classes from the most head (0) to the most tail (99), and the color intensity indicates class frequencies