SelMix: Selective Mixup Fine Tuning for Optimizing Non-Decomposable Metrics

Shrinivas Ramasubramanian[*1] **Harsh Rangwani**[*2] **Sho Takemori**[*3] **Kunal Samanta**[2]
Yuhei Umeda[3] **R. Venkatesh Babu**[2]

Abstract

Natural data often has class imbalance. This can make it difficult for machine learning models to learn to classify minority classes accurately. Industrial machine-learning applications often have objectives beyond just accuracy. For example, models may be required to meet certain fairness criteria, such as not being biased against the classes with fewer samples. These objectives are often non-decomposable in nature. SelMix is a fine-tuning technique that can be used to improve the performance of machine learning models on imbalanced data. The core idea of our framework is to determine a sampling distribution to perform a mixup of features between samples from particular classes such that it optimizes the given objective. We evaluate our technique against the existing empirical methods on standard benchmark datasets for imbalanced classification.

1. Introduction

The rise of deep networks has shown great promise by reaching near-perfect performance, particularly on tasks like visual recognition (He et al., 2022; Kolesnikov et al., 2020). It has led to their widespread deployment for practical applications, some of which also have critical consequences (Castelvecchi, 2020). Due to this, developed models must perform robustly across the entire distribution rather than only the majority part. These failure cases are often overlooked when we consider only accuracy as our primary metric for quantifying the model's performance. Therefore, more practical metrics like Recall H-Mean (Sun et al., 2006), Worst-Case (Min) Recall (Narasimhan & Menon, 2021; Mohri et al., 2019), etc., should be used for evaluation. Optimizing these metrics directly for deep networks is challenging as they cannot be expressed as a simple average of metrics calculated for each sample (Narasimhan & Menon, 2021). Optimising such metrics is termed as **Non-Decomposable Metrics (NDM)** optimization.

In prior works, there exist techniques to optimize such NDM, but their scope has mainly been restricted to linear models (Narasimhan et al., 2014; 2015a). Narasimhan & Menon (2021) in their recent work developed consistent logit-adjusted loss functions for optimizing NDM for deep neural networks. After this work in supervised setup, CSST (Rangwani et al., 2022) extends it to practical semi-supervised learning (SSL) setup, where both unlabeled and labeled data are present. Optimizing NDM presents a challenge when learning from long-tailed datasets.

In this paper, we develop a technique that utilizes the existing pre-trained classifier for representations and fine-tunes it for the desired NDM. The core contribution of our work is to develop a selective mixup sampling distribution for selecting which classes to mix up so that it optimizes the given non-decomposable objective (Figure 1). This distribution of mixup is updated periodically based on feedback from a validation set, such that it steers the model in the desired direction for optimizing the non-decomposable objective. SelMix improves the decision boundaries between particular classes to optimize the non-decomposable objective, unlike Mixup technique (Zhang et al., 2018) that applies it uniformly across all class samples. **SelMix framework can also optimize neural networks for non-linear objectives**, addressing a shortcoming of existing works (Rangwani et al., 2022; Narasimhan & Menon, 2021).

To evaluate the performance of SelMix, we perform experiments to optimize six different NDM by fine-tuning a pre-trained classifier using MiSLAS(Zhong et al., 2021) stage-1. These objectives span linear and non-linear functions of the confusion matrix. We also consider constrained optimization objectives. A list of objectives considered is available in Appendix J.3

2. Problem Setup

In our framework, we have a K-class classification problem where the data (x) comes from an instance space $\mathcal{X}$ with labels in $\mathcal{Y} := [K]$. The classifier h is a mapping where $h : \mathcal{X} \to \mathcal{Y}$. The classifier is composed of a feature extractor $g : \mathcal{X} \to \mathbb{R}^d$ which is followed by a linear softmax classification layer $f : \mathbb{R}^d \to \Delta_{K-1}$, where

[*]Equal contribution [1]Fujitsu Research of India Private Limited, Bengaluru, India [2]Vision and AI Lab, Indian Institute of Science, Bengaluru, India [3]Fujitsu Limited, Kanagawa, Japan. Correspondence to: Shrinivas <shrinivas.ramasubramanian@fujitsu.com>.

Published at the Differentiable Almost Everything Workshop of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. July 2023.

$\Delta_{K-1} \subset \mathbb{R}^K$ is the $K-1$ dimensional probability simplex. For a data sample X, the label prediction from classifier h is: $h(x) = \text{argmax}_{j\in[K]} f(g(x))_j$. For learning the classifier h we assume access to samples from data distribution $\mathcal{D}$. We denote the prior over labels as π_i , which is $\pi_i = \mathbf{P}(y = i)$. The confusion matrix for a classifier h is given as:

$$C_{ij}[h] = \mathbf{E}_{(x,y)\sim\mathcal{D}}[\mathbb{1}(y = i, h(x) = j)]. \quad (1)$$

The confusion matrix $C \in \Delta_{K\times K-1}$ and is used for characterizing the performance of a classifier. We now define ψ where $\psi : \Delta_{K\times K-1} \to \mathbb{R}$ to be the non-decomposable objective we want to optimize (or maximize). The $\psi(C[h])$ can be used for expressing a lot of practical objectives used in prior works (Cotter et al., 2019; Narasimhan et al., 2022). We elucidate some of them below, which we use in our current work. The detailed list of definitions of the variables used can be refered from Table. A.1. The worst-case recall is better suited than accuracy to evaluate classifiers on long-tailed datasets (Narasimhan & Menon, 2021) and defined as the minimum of recalls across classes $\psi^{\text{MR}}(C[h]) = \min_{1\le i\le K} \text{rec}_i[h]$, where $\text{rec}_i[h] = C_{ii}[h]/\sum_{j\in[K]} C_{ij}[h]$. Similarly, the G-mean ψ^{GM} of recalls and H-mean ψ^{HM} of recalls are defined as the geometric and the harmonic mean of recalls across classes, respectively. Fairness is another area where such complex metrics are particularly useful. Following (Cotter et al., 2019; Goh et al., 2016), we consider the maximization of the mean recall $\frac{1}{K}\sum_{i=1}^{K} \text{rec}_i[h]$ subject to $\text{cov}_i[h] \ge \alpha/K$ with a constant $\alpha > 0$. Here, $\text{cov}_i[h]$ is the predictive coverage, i.e., $\text{cov}_i[h] = \sum_{j\in[K]} C_{ij}[h]$. By using Lagrange multiplier $\boldsymbol{\lambda}$, the coverage constrained objective can be reduced to the max-min problem $\max_h \min_\lambda \psi^{\text{AM}}_{\text{cons}}(\psi)$ (See Table D.1 and (Narasimhan & Menon, 2021)).

3. Selective Mixup for Optimizing NDM

In this work, we aim to optimize the NDM using the mixup (Zhang et al., 2018) framework. Manifold Mixup (Verma et al., 2019) extends this idea to have mixups in feature space, which we use in our work. However, in mixup, the samples for mixing up are chosen randomly. This may be useful in general but can be sub-optimal when we aim to optimize for specific NDM (Table O.1). Hence, in this work, we focus on selective mixups and use them for optimizing the non-decomposable objective. The loss for mixup ($\mathcal{L}_{\text{MU}}(g(x), y, g(x'); f)$) between samples (x, y) and (x', y') having features $g(x)$ and $g(x')$ is given as:

$$\mathcal{L}_{\text{MU}} = \mathcal{L}_{\text{CE}}(f(\beta g(x) + (1-\beta)g(x')), y).$$

Here $\mathcal{L}_{\text{CE}}$ is the cross entropy loss, h is $f \circ g$ and $\beta \sim \text{Unif}(\beta_{\min}, 1), \beta_{\min} \in [0, 1)$. The above design choices follow the ideas of mixup for semi-supervised learning as used in Fan et al. (2022). We define ***(i, j) mixups*** to be the mixup of samples $(x, y) \sim D_i$ and $(x', y') \sim D_j$, where

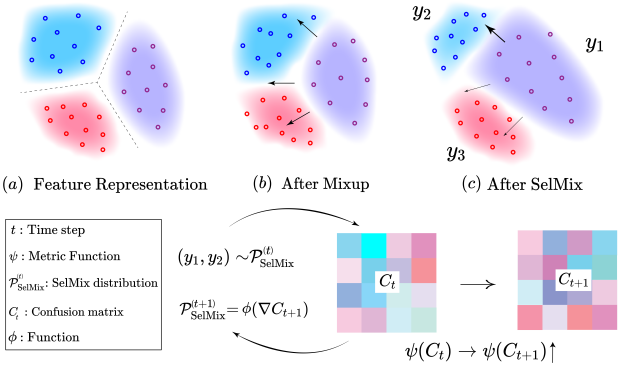


Figure 1. We demonstrate the effect of the variants of Mixup on feature representations of samples (a). With standard Mixup, the feature representation gets equal contribution in all directions of other classes (b). For SelMix (c), Mixups for specific classes are selected such that they optimize the desired metric.

D_i is defined as the set of samples with class label i i.e., $\{(x, y) : y = i\}$. Although our main focus is supervised learning, in the semi-supervised setting, we consider an unlabeled sample x' with a pseudo label y' and replace D_j by the set of samples $\widetilde{D}_j$ with pseudo label j. Even in the semi-supervised setting, the set D_i remains to be the set of samples (x, y) with true label i. In our method, we assume access to a held-out validation dataset D^{val}. For analyzing the effect of (i, j) mixups on the model, we use the loss incurred by mixing the centroids of class samples given as $z_k = \mathbb{E}_{x\sim D^{\text{val}}_k}[g(x)]$ for each class k. This loss which is representative of the expected loss due to (i, j) mixup is:

$$\bar{\mathcal{L}}^{i,j}_{\text{MU}} = \mathcal{L}_{\text{MU}}(z_i, i, z_j; f) \quad \forall i, j \in [K] \times [K]. \quad (2)$$

In our work, we mainly focus on analysis of linear classifier f having weight ν as we primarily train that for optimizing the NDM. We use the gradient of the above representative loss to characterize the change in linear classifier weights (ΔW) due to (i, j) mixup, given as $\nu_{ij} = -\eta\nabla\bar{\mathcal{L}}^{ij}_{\text{MU-CE}}$. These $\nu_{i,j}$ directions characterize the average weight change due to (i, j) mixup.

Our framework is based on the idea of optimizing the linear approximation of non-decomposable objective $\psi(C)$ in terms of directional derivative, in the directions of change in weights (ν_{ij}) due to (i, j) mixups. This differs from earlier Cost Sensitive Learning (CSL) frameworks for non-decomposable objective optimization (Narasimhan & Menon, 2021; Cotter et al., 2019; Narasimhan et al., 2015b;a), as they consider linear approximation w.r.t. the entries of confusion matrix. Hence, our work gives a new orthogonal direction to the problem of non-decomposable objective optimization. We define gain $\mathbf{G}$ matrix as the expected rate of increase in metric $\psi(C)$ upon performing (i, j) mixup and performing a step of stochastic gradient descent. It is given as the directional derivative of $\psi(C)$ w.r.t the parameters along the direction of update of weights

for a given $\bar{\mathcal{L}}^{ij}_{\text{MU}}$:

$$G_{ij} = \nabla_{\nu_{ij}} \psi(C) \quad \text{where } \nu_{ij} = -\eta \nabla \bar{\mathcal{L}}^{ij}_{\text{MU}}. \tag{3}$$

We now define a distribution $\mathcal{P}_{\text{Mix}}(i,j)$ over the labels i's and pseudo-labels j's of the samples that shall be mixed up. As we aim to maximize the average gain $\mathbb{E}_{(i,j)}[G_{ij}]$, one greedy strategy could be to only mixup (i,j) pairs corresponding to $\max_{i,j} G_{ij}$. However that strategy doesn't work as we do n mixups and in that the linear approximation becomes invalid (Refer Table O.2 for evidence). Hence, we select the $\mathcal{P}_{\text{Mix}}$ to be the scaled softmax of the gain matrix as our strategy, with $s > 0$ given as ($\mathcal{P}_{\text{SelMix}}$): $\mathcal{P}_{\text{SelMix}}(i,j) = \text{softmax}(sG_{ij})$. We provide theoretical results regarding the optimality of the proposed $\mathcal{P}_{\text{SelMix}}$ in Sec. F.

We now move to the second crucial practical component required for the algorithm: the tractable gain estimation (G_{ij}) for each (i,j) mixup. However, while trying to optimize the non-decomposable objective, it's required that the confusion matrix $C[h]$ should satisfy the following constraints: $\sum_j C_{i,j} = \pi_i$, $\sum_{i,j} C_{i,j} = 1$ with $0 \leq C_{i,j} \leq 1$. For implicitly enforcing these constraints we make use of the re-parameterization (Achiam et al., 2017) trick to define the row of confusion matrix (C_i) in terms of the row of unconstrained matrix $\tilde{C} \in \mathbb{R}^{K \times K}$. We define $C_i := [\pi_i \cdot \text{softmax}(\tilde{C}_i)]$. The gain can be analagously defined (Eq. 3) in terms of $\tilde{C}$: $G_{ij} = \nabla_{\nu_{ij}} \psi(\tilde{C}) \quad \text{where } \nu_{ij} = -\eta \nabla \bar{\mathcal{L}}^{ij}_{\text{MU}}$.

We want to convey that despite the re-parameterization we do not require the actual computation of $\tilde{C}$. As all the terms of $\frac{\partial \psi(\tilde{C})}{\partial \tilde{C}_{lk}}$ which we require, can be computed analytically in terms of C, which makes this operation inexpensive (see Sec. B). Further, this also allows us to have a tractable formula for gain circumeventing the non-differentiability introduced by constraints of confusion matrix C. (see Sec. H) We now provide an approximation for gain through the following. Here for conciseness, we provide an informal statement of the approximation formula. We refer to Theorem G.1 for a more mathematically precise statement.

Theorem 3.1. *The Gain for the i,j mixup (G_{ij}) can be approximated as $G_{ij} \approx \sum_{k,l} \frac{\partial \psi(\tilde{C})}{\partial \tilde{C}_{kl}} \left((\nu_{ij})_l^\top \cdot z_k \right)$. where $z_k = \mathbb{E}_{x \sim D_k^{\text{val}}}[g(x)]$ is the mean of the feature of the samples from validation set belonging to class k.*

The above formula is an approximation based on the correlation of change in logits for the classifier f with weight W, with the change in the unconstrained version of the confusion matrix. Changes in the confusion matrix entries are necessary to be correlated with changes in logit vectors. This approximation formula holds under natural conditions (Theorem G.1). Moreover, this approximation works well in practice, as demonstrated empirically in Sec. 4.

We now provide an algorithm for training h through SelMix (We refer to Alg. 1 for psuedo code). The high-level idea is to perform training cycles, in each of which you estimate the gain matrix $\mathbf{G}$ through a validation set and use it for training the neural network for a few Stochastic Gradient Descent (SGD) steps. As our expressions of gain are based on linear classifier, we only train linear classifier fully and fine-tune the backbone slightly for better empirical results (Sec. 4). For expressing the algorithm formally, we introduce the time-dependent notations for gain ($G^{(t)}_{ij}$), the classifier $h^{(t)}$, the SelMix distribution $\mathcal{P}^{(t)}_{\text{SelMix}}$, weight-direction change $\nu^{(t)}_{ij}$. At each iteration $t = 1, \ldots, T$, we compute the SelMix distribution $\mathcal{P}^{(t)}_{\text{SelMix}} = \text{softmax}(s\mathbf{G}^{(t)})$ using Thm.3.1. We randomly sample class labels Y_1 and Y_2 by $Y_1, Y_2 \sim \mathcal{P}^{(t)}_{\text{SelMix}}$, and sample (X_1, Y_1) and (X_2, Y_2) uniformly random from D_{Y_1} and D_{Y_2} respectively. Then, we update the classifier $h^{(t)}$ by using the CE and the mixup sample, i.e., $h^{(t+1)} := \text{SGD-Update}(h^{(t)}, \nabla \mathcal{L}_{\text{MU}}(g(X_1), Y_1, g(X_2)))$. In each iteration t, we repeat this SGD update n times.

4. Experimental Analysis

4.1. Experimental Setup

We follow the convention for long-tailed classification where the classes are indexed $1...K$ with the number of samples per class decreasing as the class index increases. The data distribution is exponentially decreasing in nature. The number of samples in class i are denoted as N_i. The severity of imbalance is quantified by $\rho = N_1/N_K$.

We show the efficacy of SelMix for optimizing a wide variety of NDM for CIFAR 10,100 LT ($\rho = 100$) and Imagenet-1k LT datasets . Our classifier h is composed of a feature extractor $g : \mathcal{X} \to \mathbb{R}^d$ followed by a linear softmax classification layer $f : \mathbb{R}^d \to \Delta_{K-1}$, as mentioned in Sec. 2. As our initial classifier we use ResNet (He et al., 2016) pre-trained using the first stage of MiSLAS (Zhong et al., 2021). We perform fine-tuning of the model through SelMix (Alg. 1) that generates a distinct sampling function to perform mixup that specifically optimised the desired objective. The appendix mentions additional details and hyper-parameters values in Table J.1.

Evaluation Setup. We compare the methods on two broad sets of metric objectives: **a) Unconstrained objectives** which includes G-mean, H-mean, Mean (Arithmetic Mean), and worst-case (Min.) Recall **b) Constrained objectives** include maximizing the recalls under coverage constraints. The constraint for all classes is that coverage should be greater than $\frac{0.95}{K}$. As followed in the literature (Narasimhan & Menon, 2021) for CIFAR-100 LT, Imagenet-100 LT, and Imagenet-1k LT due to very few samples in the tail classes, instead of Min Recall/Coverage, we optimize the Min Head-Tail Recall/Min Head-Tail coverage, respectively. The tail corresponds to the least frequent 10% of the classes, and the head corresponds to the rest. For a more detailed overview of the metric objectives and their definition (Table J.3).

Table 1. Comparison of metric values with various Long-Tailed methods on CIFAR-10/100 LT under $\rho = 100$ setup. The best results are indicated in bold. We consistently observe a boost in the performance compared to the initial pre-trained model of MiSLAS (stage 1)

		Mean Rec.	Min Rec.	GM	HM	Mean Rec. / Min Cov.		Mean Rec.	Min H-T Rec.	HM	GM	Mean Rec. / Min HT Cov.
ERM	CIFAR-10	70.1	40.2	66.9	63.7	70.1/0.041	CIFAR-100	41.7	10.5	31.2	19.0	41.7/0.0029
LDAM w/ DRW		76.3	60.9	75.4	74.5	76.3/0.066		42.6	10.2	33.2	19.3	42.6/0.0047
CSL		79.7	71.1	79.5	79.2	79.7/0.091		41.9	36.2	26.4	12.2	41.9/0.0990
MiSLAS (stage 1)		74.9	45.2	72.7	70.3	74.9/0.046		40.2	1.1	0.0	0.0	40.2/0.0020
w/ (stage 2)		81.9	72.5	81.6	81.3	81.9/0.077		47.0	15.2	39.9	30.9	47.0/0.0055
w/ SelMix		**83.3**	**79.2**	**82.8**	**82.6**	**83.2/0.095**		**48.3**	**41.3**	**42.3**	**38.2**	**47.8/0.0095**

4.2. Comparison

CIFAR datasets: Here, we consider the CIFAR-10 and CIFAR-100 LT ($\rho = 100$) datasets. We observe a significant improvement in the corresponding metric for which the model was fine-tuned when using the proposed SelMix method in all cases, compared to the FixMatch (LA) pre-trained model. For the Min. Recall metric, we observe an approximately 75% improvement for CIFAR-10 LT and a corresponding 48% improvement in Min. HT Recall for CIFAR-100 LT over the baseline method. SelMix also outperforms the 2nd stage of MiSLAS across all metrics, including mean recall ($\sim$ accuracy). We find that, for optimizing a given metric, $\mathcal{P}_{\text{SelMix}}$ initially mixes up samples from tail classes with head classes while using the tail class sample's label to increase performance on them. Then, it gradually transitions towards uniform mixups later (Appendix L).

Consider the optimization of metrics with coverage constraints (i.e., $\text{cov}_i[h] \geq \frac{0.95}{K}$). We optimize the model's mean recall with coverage constraint, as CSL(Narasimhan & Menon, 2021) supports it. However, as SelMix is generic, it also supports optimizing non-linear metrics like H-mean with coverage, which we show in the Appendix K. Table 1 shows that most heuristic SotA methods lead to sub-optimal min. coverage values, and only CSL and SelMix approximately satisfy the coverage constraints. SelMix achieves better mean recall than CSL, providing a better tradeoff in terms of performance and satisfiability of constraints.

Table 2. Comparison of SelMix's performance on Imagnet-1k LT. We show the scalability of SelMix to large scale datasets with very minimal cost of fine-tuning the pre-trained model

Method	Mean *Recall*	Min *Recall*	Mean Rec. / Min HT Cov.
CSL	48.5	40.2	38.5/0.00099
MiSLAS (stage 1)	45.5	4.1	45.5/0.00004
w/ (stage 2)	52.2	29.7	52.2/0.00062
w/ SelMix	**52.8**	**45.1**	**52.5/0.00099**

Imagenet1k LT We show that our method SelMix, scales well to large-scale datasets as well. Here we do not consider the Harmonic mean of recall and Geometric mean of recall since due to the scarcity of samples for the tail classes (5 samples per class), their recall values are very small pushing the overall values too low (≈ 0) making them not very informative about the overall performance of the classifier. Here again, we observe that SelMix significantly improves over the baseline and also comfortably satisfies the coverage constraints without suffering on the mean recall. This is unlike Stage-2 MiSLAS trained model which does not satisfy the coverage constraint.

4.3. Extension to Semi-Supervised Learning

We show results for the case of a semi-supervised setup where the samples are taken from the labeled and unlabeled datasets based on their labels and pseudo-labels, respectively. We compare against existing SoTA methods in imbalanced semi-supervised learning such as DASO (Oh et al., 2022), CoSSL (Fan et al., 2022), and ABC (Lee et al., 2021). We show that and our method not only achieves superior performance for the desired metric but also achieves superior accuracy ($\sim$ Mean Rec.) under a diverse set of data distributions, even under cases where the labeled and unlabeled data distribution are mismatched. For the metric of Min Recall, a 5% improvement is observed for CIFAR-10, and a corresponding 9.8% improvement in Min HT Recall for CIFAR-100 over existing SoTA methods (Table O.5).

The STL-10 dataset comes with an additional 100k samples, with unknown label distribution. This setting emulates the practical scenario where a lot of data is being collected but labels are absent due to high annotation costs. Due to no distributional assumption, SelMix outperforms SoTA methods for the min-recall metric by 12.7% (Table O.3).

We use a WideResNet 28-2 pre-trained using FixMatch (Sohn et al., 2020) where the supervised loss is replaced by the logit adjusted loss for superior consistency regularization despite the label distribution mismatch between labeled and unlabeled data. Training details are available in Table J.2.

5. Conclusion and Discussion

We study the optimization of complex practical metrics like the G-mean and H-mean of Recalls, along with objectives with fairness constraints in the case of neural networks. We find that most existing techniques achieve sub-optimal performance in terms of these practical metrics, notably on worst-case recall. These metrics and constraints are NDM, for which we propose a Selective Mixup (SelMix) based fine-tuning algorithm for optimizing them. The algorithm selects samples from particular classes to mixup to improve a linear approximation of the non-decomposable objective. Our method SelMix is able to improve on the majority of objectives in comparison to the baselines, bridging the gap between theory and practice. We expect SelMix fine-tuning technique to be used for improving existing models by improving on worst-case and fairness metrics inexpensively.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

Auer, P., Cesa-Bianch, N., Freund, Y., and E. Schapire, R. The non-stochastic multi-armed bandit problem. *SIAM Journal of Computing*, 32(1):48–77, 2002.

Castelvecchi, D. Is facial recognition too biased to be let loose? *Nature*, 587(7834):347–350, 2020.

Cotter, A., Jiang, H., Gupta, M. R., Wang, S., Narayan, T., You, S., and Sridharan, K. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *J. Mach. Learn. Res.*, 20 (172):1–59, 2019.

Fan, Y., Dai, D., Kukleva, A., and Schiele, B. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Goh, G., Cotter, A., Gupta, M., and Friedlander, M. P. Satisfying real-world goals with dataset constraints. *Advances in Neural Information Processing Systems*, 29, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Kennedy, K., Mac Namee, B., and Delany, S. J. Learning without default: A study of one-class classification and the low-default portfolio problem. In *Artificial Intelligence and Cognitive Science: 20th Irish Conference, AICS 2009, Dublin, Ireland, August 19-21, 2009, Revised Selected Papers 20*, pp. 174–187. Springer, 2010.

Kim, J., Hur, Y., Park, S., Yang, E., Hwang, S. J., and Shin, J. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020a. Curran Associates Inc. ISBN 9781713829546.

Kim, J.-H., Choo, W., and Song, H. O. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pp. 5275–5285. PMLR, 2020b.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pp. 491–507. Springer, 2020.

Lattimore, T. and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

Lee, H., Shin, S., and Kim, H. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems*, 34: 7082–7094, 2021.

Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2020.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Narasimhan, H. and Menon, A. K. Training over-parameterized models with non-decomposable objectives. *Advances in Neural Information Processing Systems*, 34, 2021.

Narasimhan, H., Vaish, R., and Agarwal, S. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. *Advances in neural information processing systems*, 27, 2014.

Narasimhan, H., Kar, P., and Jain, P. Optimizing non-decomposable performance measures: A tale of two classes. In *International Conference on Machine Learning*, pp. 199–208. PMLR, 2015a.

Narasimhan, H., Ramaswamy, H., Saha, A., and Agarwal, S. Consistent multiclass algorithms for complex performance measures. In *International Conference on Machine Learning*, pp. 2398–2407. PMLR, 2015b.

Narasimhan, H., Ramaswamy, H. G., Tavker, S. K., Khurana, D., Netrapalli, P., and Agarwal, S. Consistent multiclass algorithms for complex metrics and constraints. *arXiv preprint arXiv:2210.09695*, 2022.

Oh, Y., Kim, D.-J., and Kweon, I. S. Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9786–9796, 2022.

Rangwani, H., Ramasubramanian, S., Takemori, S., Takashi, K., Umeda, Y., and Radhakrishnan, V. B. Cost-sensitive self-training for optimizing non-decomposable metrics. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

Sun, Y., Kamel, M. S., and Wang, Y. Boosting for learning multiple classes with imbalanced class distribution. In *Sixth international conference on data mining (ICDM'06)*, pp. 592–602. IEEE, 2006.

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pp. 6438–6447. PMLR, 2019.

Wang, S. and Yao, X. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4): 1119–1130, 2012.

Wei, C., Sohn, K., Mellina, C., Yuille, A., and Yang, F. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10857–10866, 2021.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Zhong, Z., Cui, J., Liu, S., and Jia, J. Improving calibration for long-tailed recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16489–16498, 2021.

Appendix

A. Notation

Table A.1. Table of Notations used in Paper

K	Number of classes
$\mathcal{Y} := [K]$	Label space
x	Instance
y	Label
$\mathcal{X}$	Instance space
$h : \mathcal{X} \to \mathcal{Y}$	a classifier
$C[h]$	Confusion matrix for the classifier h
$\Delta_{n-1} \subset \mathbb{R}^n$	the $n-1$-dimensional probability simplex
$\psi : \Delta_{K^2-1} (\subset \mathbb{R}^{K \times K}) \to \mathbb{R}$	a function defined on the set of confusion matrices ($\psi(C[h])$ is the metric of h)
π_i	prior for class $i \in [K]$
$g : \mathcal{X} \to \mathbb{R}^d$	a feature extractor
$f : \mathbb{R}^d \to \Delta_{K-1}$	the final classifier such that $h = \mathrm{argmax}_i \, f_i \circ g$
$W \in \mathbb{R}^{K \times d}$	the weight of the final layer
z_k	the centroid of class samples given as $\mathbb{E}_{x \sim D_k^{\mathrm{val}}} [g(x)]$
$\mathcal{L}_{\mathrm{MU}}(x_i, y_i, x_j; h)$	the loss for mixup between labeled sample (x_i, y_i) and unlabeld sample x_j
$\overline{\mathcal{L}}_{\mathrm{MU}}^{i,j}$	the expected loss due to (i, j) mixup
G_{ij}	gain upon performing (i, j) mixup
$\widetilde{\nu}_{ij}$	the change in linear classifier weights ν due to (i, j) mixup
$\nabla_A \psi$ (where $A \in \mathbb{R}^{K \times d}$)	the directional derivative defined as $\sum_{k,l} A_{kl} \frac{\partial \psi}{\partial W_{kl}}$
s	the inverse temperature parameter for the softmax
$\tilde{C}$	Unconstrained extension for confusion matrix C
D_i	Subset of data with label i
$\tilde{D}_i$	Subset of data with pseudo-label i
$\mathcal{P}$	a distribution on $[K] \times [K]$
$\boldsymbol{\mathcal{P}} = (\mathcal{P}^t)_{t=1}^T$	a policy (a sequence of distributions $\mathcal{P}^t$)
$\overline{G}(\boldsymbol{\mathcal{P}})$	the expected average gain of $\boldsymbol{\mathcal{P}}$
N_k	the number of samples in the k-th labeled class
M_k	the number of samples in the k-th unlabeled class
ρ_l	the class imbalanced factor of the labeled dataset ($\max_{1 \le i,j \le K} N_i / N_j$)
ρ_u	the class imbalanced factor of the unlabeled dataset
$\mathcal{H}$	The set of first 90% classes that contains the majority of samples
$\mathcal{T}$	The set of last 10% classes that contains the minority of samples
$\|A\|_{\mathrm{F}}$	the Frobenius norm of a matrix

B. Non-differentiability and the Unconstrained Confusion Matrix

In Eq. (3), we define the gain G_{ij} by a directional derivative of $\psi(C)$ with respect to weight W. However, strictly speaking, since the definition of the confusion matrix C involves the indicator function, $\psi(C)$ is not a differentiable function of W. Moreover, even if gradients are defined, they vanish because of the definition of the indicator function. In the assumption of Theorem G.1 (a formal version of Theorem 3.1), we assume $\widetilde{C}$ is a smooth function of W and it implies C is a differentiable function of W. This assumption can be satisfied if we replace the indicator function by surrogate functions of the indicator functions in the definition of the confusion matrix C. More precisely, we replace the definition of $C_{ij}[h] = \pi_i \mathbb{E}_{x \sim P_i} [\mathbb{1}(h(x) = j)]$ by $\pi_i \mathbb{E}_{x \sim P_i} [s_j(f(x))]$. Here $h(x) = \mathrm{argmax}_k \, f_k(x)$ as before, P_i is the class conditional distribution $P(x|y = i)$ and s_j is a surrogate function of $p \mapsto \mathbb{1}(\mathrm{argmax}_i \, p_i = j)$ satisfying $0 \le s_j(p) \le 1$

for any $1 \leq j \leq K$, $p \in \Delta_{K-1}$ and $\sum_{j=1}^{K} s_j(p) = 1$ for any $p \in \Delta_{K-1}$. To compute G_{ij}, one can directly use the definition of Eq. (3) with the smoothed confusion matrix using surrogate functions of the indicator function. However, an optimal choice of the surrogate function is unknown. Therefore, in this paper, we introduce an unconstrained confusion matrix $\tilde{C}$ and the approximation formula Theorem 3.1 (Theorem G.1). An advantage of introducing $\widetilde{C}$ and the approximation formula is that the RHS of the approximation formula $\sum_{k,l} \frac{\partial \psi(\tilde{C})}{\partial \tilde{C}_{kl}} \left((\nu_{ij})_l^\top \cdot z_k\right)$ does not depend on the choice of the surrogate function if we use formulas provided in Sec. H with the original (non-differentiable) definition of C. [1]. Since the optimal choice of the surrogate function is unknown, this gives a reliable approximation.

C. Computational Complexity

We discuss the computational complexity of SelMix and that of an existing method (Rangwani et al., 2022) for NDM optimization in terms of the class number K. We note that to the best of our knowledge, CSST (Rangwani et al., 2022) is the only existing method for NDM optimization in the SSL setting.

Proposition C.1. *The following statements hold:*

1. *In each iteration t in Algorithm 1, computational complexity for $\mathcal{P}^{(t)}_{\text{SelMix}}$ is given as $O(K^3)$.*
2. *In each iteration of CSST (Rangwani et al., 2022), it needs procedure that takes $O(K^3)$ time.*

Here, the Big-O notation hides sizes of parameters of the network other than K (i.e., the number of rows of W) and the size of the validation dataset.

Proof. **1.** Computational complexity for the confusion matrix is given as $O(K^3)$ since there are K^2 entries and for each entry, evaluating $h^{(t)}(x)$ takes $O(K)$ time for each validation data x. For each $1 \leq k \leq K$, computational complexity for z_k is $O(K)$. We compute $\{\text{softmax}(z_k)\}_{1 \leq k \leq K}$, which takes $O(K^2)$ time. The (m, l)-th entry of the matrix ν_{ij} is given as $-\eta \zeta_m(\delta_{il} - \text{softmax}_i(\zeta))$, where $1 \leq m \leq d$, $1 \leq l \leq K$, and $\zeta = \beta z_i + (1-\beta) z_j \in \mathbb{R}^d$. Therefore, once we compute $\{\text{softmax}(z_k)\}_{1 \leq k \leq K}$, computational complexity for $\{\nu_{ij}\}_{1 \leq i,j \leq K}$ is $O(K^3)$. For each $1 \leq l \leq K$, we put $v_l = \sum_{k=1}^{K} \frac{\partial \psi(C^{(t)})}{\partial \tilde{C}_{kl}} z_k$. Then computational complexity for $\{v_l\}_{1 \leq l \leq K}$ is $O(K^2)$. Since $G^{(t)}_{ij} = \sum_{l=1}^{K} (\nu^{(t)}_{ij})_l^\top \cdot v_l$ is a sum of K dot products of d-dimensional vectors, once we compute $\{v_l\}_l$, computational complexity for $\{G^{(t)}_{ij}\}_{1 \leq i,j \leq K}$ is $O(K^3)$. Thus, computational complexity for $\mathcal{P}^{(t)}_{\text{SelMix}}$ is given as $O(K^3)$.

2. In each iteration t, CSST needs computation of a confusion matrix at validation dataset. Since there are K^2 entries and for each entry, $h^{(t)}(x)$ takes $O(K)$ time for each validation data x, computational complexity for the confusion matrix is given as $O(K^3)$. Thus, we have our assertion. □

D. Non-Decomposable Metrics

In this section, we provide more detailed introduction to NDM. Real-world datasets are long-tailed and imbalanced. In such cases, the mean recall can be deceptive as the model might be very good for majority classes while performing below par for minority classes. In such cases, the metrics of H-mean (the harmonic mean of recalls across classes) (Kennedy et al., 2010), G-mean (the geometric mean of recalls across classes) (Wang & Yao, 2012; Lee et al., 2021) and Minimum (worst-case) recall (the minimum of recalls across classes) (Narasimhan & Menon, 2021) across classes is better suited for holistic evaluation. These metrics show a significant deviation from mean performance in case of performance disparity between the majority and minority classes. The G-mean of recall can be defined in terms of the confusion matrix ($C[h]$) as $\psi^{\text{GM}}(C[h]) = \left(\prod_{i \in [K]} \frac{C_{ii}[h]}{\sum_{j \in [K]} C_{ij}[h]}\right)^{\frac{1}{K}}$. For the minimum recall (ψ^{MR}) we use the continuous relaxation as used by (Narasimhan & Menon, 2021), by writing the objective as min-max optimization over $\boldsymbol{\lambda} \in \Delta_{K-1}$: $\max_h \psi^{\text{MR}}(C[h]) = \max_h \min_{\boldsymbol{\lambda} \in \Delta_{K-1}} \sum_{i \in [K]} \lambda_i \frac{C_{ii}[h]}{\sum_{j \in [K]} C_{ij}[h]}$. Fairness is another area where such complex metrics are particularly useful. For example, prior works (Cotter et al., 2019; Goh et al., 2016) consider optimizing the mean recall while constraining the predictive coverage ($\text{cov}_i[C[h]] = \sum_j C_{ji}$) that is the proportion of class i predictions on test data given as $\max_h \frac{1}{K} \sum_{i=1}^{K} rec_i[h]$ s.t. $\text{cov}_i[h] \geq \frac{\alpha}{K}$ $\forall i \in [K]$. Optimization of above-constrained objectives is possible by

[1] Constants such as c, c' in Theorem G.1 do depend on the surrogate function

Table D.1. Metrics defined using the entries of confusion matrix C.

Metric	Definition
Mean Recall (ψ^{AM})	$\frac{1}{K}\sum_{i\in[K]}\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}$
Min. Recall (ψ^{MR})	$\min_{\boldsymbol{\lambda}\in\Delta_{K-1}}\sum_{i\in[K]}\lambda_i\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}$
G-mean (ψ^{GM})	$\left(\prod_{i\in[K]}\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}\right)^{\frac{1}{K}}$
H-mean (ψ^{HM})	$K\left(\sum_{i\in[K]}\frac{\sum_{j\in[K]}C_{ij}[h]}{C_{ii}[h]}\right)^{-1}$
Mean Recall s.t. per class coverage $\geq\tau$ ($\psi^{\text{AM}}_{\text{cons.}}$)	$\min_{\boldsymbol{\lambda}\in\mathbb{R}^K_+}\sum_{i\in[K]}\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}+\sum_{j\in[K]}\lambda_j\left(\sum_{i\in[K]}C_{ij}[h]-\frac{\alpha}{K}\right)$

using the Lagrange Multipliers ($\boldsymbol{\lambda}\in\mathbb{R}^K_{\geq 0}$) as done in Sec. 2 of Narasimhan & Menon (2021). By expressing this above expression in terms of $C[h]$ and through linear approximation, the constrained objective $\psi_{\text{cons.}}(C[h])$ can be considered as: $\max_h \psi^{\text{AM}}_{\text{cons.}}(C[h]) = \max_h \min_{\boldsymbol{\lambda}\in\mathbb{R}^K_{\geq 0}} \frac{1}{K}\sum_{i\in[K]}C_{ii}[h]/\sum_{j\in[K]}C_{ij}[h]+\sum_{j\in[K]}\lambda_j\left(\sum_{i\in[K]}C_{ij}[h]-\frac{\alpha}{K}\right)$.

The $\boldsymbol{\lambda}$ for calculating value of $\psi_{\text{cons.}}(C[h])$ and $\psi^{\text{MR}}(C[h])$, is periodically updated using exponentiated or projected gradient descent as done in (Narasimhan & Menon, 2021). We summarize $\psi(C[h])$ for all NDM we consider in this paper in Table D.1. We will consider maximization of all such objectives expressed through $\psi(C[h])$. In comparison to existing frameworks (Narasimhan & Menon, 2021; Rangwani et al., 2022) in addition to linear metrics, we can also optimize for non-linear metrics like (G-mean and H-mean) for neural networks. , such as optimizing minimum Recall (Narasimhan & Menon, 2021; Cotter et al., 2019), H-mean of Recall and even the lagrangian relaxations of constrained objectives like mean recall under coverage constraints. We follow the convention that increasing ψ leads to the desired performance. Givena K class classification problem, our objective is to optimize a NDM $\psi(C)$, where $\psi:\Delta_{K\times K-1}\rightarrow\mathbb{R}$, which is a function of the entries of the confusion matrix $C[h]$, for a classifier $h\in\mathcal{H}$. Here $\mathcal{H}$ is the set of all possible classifiers. Some of the NDM we wish to optimise for have been tabulated below. Many of these metrics encourage the classifier to produce more equitable results on long-tailed data. We assume a feature extractor $g:\chi\rightarrow\mathbb{R}^d$ which is followed by a linear layer classifier f parameterised by weights $W\in\mathbb{R}^{d\times K}$ which is followed by a softmax layer.

E. Algorithm

In this section, we provide more detailed description of our algorithm in Algorithm 1.

Algorithm 1 Training through SelMix

Input: Data (D, D^{val}), iterations T, classifier $h^{(0)}$, metric function ψ
for $t=1$ **to** T **do**
  $h^{(t)}=h^{(t-1)}$, $C^{(t)}=\mathbb{E}_{(x,y)\sim D^{\text{val}}}[C[h^{(t)}]]$
  $\nu^{(t)}_{ij}=-\eta\nabla\tilde{\mathcal{L}}^{ij}_{\text{MU-CE}}\quad\forall\, i,j$ (3)
  $G^{(t)}_{ij}=\sum_{k,l}\frac{\partial\psi(C^{(t)})}{\partial\tilde{C}_{kl}}(\nu^{(t)}_{ij})^{\top}_{l}\cdot z_k\quad\forall\, i,j$
  $\mathcal{P}^{(t)}_{\text{SelMix}}=\text{softmax}(\mathbf{G}^{(t)})$
  for n SGD steps **do**
    $Y_1,Y_2\sim\mathcal{P}^{(t)}_{\text{SelMix}}$
    $X_1\sim\mathcal{U}(D_{Y_1})\,,\ X_2\sim\mathcal{U}(D_{Y_2})$ *// sample batches of data*
    $h^{(t)}:=\text{SGD-Update}(h^{(t)},\nabla\mathcal{L}_{\text{MU-CE}}(X_1,Y_1,X_2))$
  end for
end for
Output: $h^{(T)}$

F. Theoretical Analysis

In this section motivated by Algorithm 1, we consider the following online learning problem and prove validity of our method. For each time step $t = 1, \dots, T$, an agent selects pairs $(i^{(t)}, j^{(t)}) \in [K] \times [K]$, where random variables $(i^{(t)}, j^{(t)})$ follows a distribution $\mathcal{P}^t$ on $[K] \times [K]$. We call a sequence of distributions $(\mathcal{P}^t)_{t=1}^T$ a policy. For $(i, j) \in [K] \times [K]$ and $1 \leq t \leq T$, we assume that a random variable $G_{ij}^{(t)}$ is defined. We regard $G_{ij}^{(t)}$ as the gain in the metric when performing (i, j)-mixup at iteration t in Algorithm 1. We assume that $G_{ij}^{(t)}$ is a random variable due to randomness of the validation dataset, X_1, X_2, and the policy. Furthermore, we assume that when selecting $(i^{(t)}, j^{(t)})$, the agent observes random variables $G_{ij}^{(t)}$ for $(i, j) \in [K] \times [K]$ but cannot observe the true gain defined by $\mathbb{E}\left[G_{ij}^{(t)}\right]$. The average gain $\overline{G}^{(T)}(\boldsymbol{\mathcal{P}})$ of a policy $\boldsymbol{\mathcal{P}} = (\mathcal{P}^t)_{t=1}^T$ is defined as $\overline{G}^{(T)}(\boldsymbol{\mathcal{P}}) = \frac{1}{T}\sum_{t=1}^T \mathbb{E}\left[G_{ij}^{(t)}\right]$, where $(i^{(t)}, j^{(t)})$ follows the distribution $\mathcal{P}^t$ and the expectation is taken with respect to the randomness of the policy, validation dataset, X_1, X_2. This problem setting is similar to that of Hedge (Freund & Schapire, 1997) and EXP3 (Auer et al., 2002). However, in the problem setting of Hedge, the agent observes gains (or losses) after performing an action but in our problem setting, the agent have random estimations of the gains before performing an action. We note that even in this setting, methods such as argmax with respect to $G_{ij}^{(t)}$ may not perform well due to randomness of $G_{ij}^{(t)}$ and errors in the approximation.

We call a policy $\boldsymbol{\mathcal{P}} = (\mathcal{P}^t)_{t=1}^T$ non-adaptive (or stationary) if $\mathcal{P}^t$ is the same for all $t = 1, \dots, T$, i.e, if there exists a distribution $\mathcal{P}^0$ on $[K] \times [K]$ such that $\mathcal{P}^t = \mathcal{P}^0$ for all $t = 1, \dots, T$. A typical example of non-adaptive policies is the uniform mixup, i.e., $\mathcal{P}^t$ is the uniform distribution on $[K] \times [K]$. Another typical example is $\mathcal{P}^t = \delta_{(i^{(0)}, j^{(0)})}$ for a fixed $(i^{(0)}, j^{(0)}) \in [K] \times [K]$ (i.e., the agent performs the fixed $(i^{(0)}, j^{(0)})$)-mixup in each iteration). Similarly to Hedge (Freund & Schapire, 1997) and EXP3 (Auer et al., 2002), we consider softmax policy with respect to the cumulative sum of gains.

Similary to Hedge (Freund & Schapire, 1997), the following theorem states that the SelMix policy is better than any non-adaptive policy in terms of the average expected gain if T is sufficiently large:

Theorem F.1. *We define $\boldsymbol{\mathcal{P}}_{\textit{SelMix}} = (\mathcal{P}^t_{\textit{SelMix}})_{t=1}^T$ by $\mathcal{P}^t_{\textit{SelMix}} = \mathrm{softmax}((s\sum_{\tau=1}^{t-1} G_{ij}^{(\tau)})_{1 \leq i,j \leq K})$, where $s > 0$ is the inverse temperature parameter. We assume that $G_{ij}^{(t)}$ is normalized so that $G_{ij}^{(t)} \in [0, 1]$. Then, with an appropriate choice of the inverse temperature parameter s, for any non-adaptive policy $\boldsymbol{\mathcal{P}}^0 = (\mathcal{P}^0)_{t=1}^T$, we have (Proof in Appendix G.2)*

$$\overline{G}^{(T)}(\boldsymbol{\mathcal{P}}_{\textit{SelMix}}) + \frac{2\sqrt{\log K}}{\sqrt{T}} \geq \overline{G}^{(T)}(\boldsymbol{\mathcal{P}}^0).$$

Next, we consider a variant of the policy and introduce analysis of it.

Theorem F.2. *We define $\mathcal{P}^{(t)}_{\mathrm{SelMix}} = \mathrm{softmax}((s\sum_{\tau=1}^{t} G_{ij}^{(\tau)})_{ij})$ and define a policy $\boldsymbol{\mathcal{P}}_{\mathrm{SelMix}} = (\mathcal{P}^{(t)}_{\mathrm{SelMix}})_t$. Then for any $s > 0$ and for any non-adaptive policy $\boldsymbol{\mathcal{P}}^{(0)} = (\mathcal{P}^{(0)})_{t=1}^T$, we have*

$$\overline{G}^{(T)}(\boldsymbol{\mathcal{P}}_{\textit{SelMix}}) + \frac{2\log K}{sT} \geq \overline{G}^{(T)}(\boldsymbol{\mathcal{P}}^{(0)}).$$

G. Proofs for Mathematical Results in Paper

G.1. A Formal Statement of Theorem 3.1 and its Proof

We provide a more formal statement of Theorem 3.1 (Theorem G.1) and provide its proof.

Theorem G.1. *For a matrix $A \in \mathbb{R}^{n \times m}$, we denote by $\|A\|_{\mathrm{F}}$ the Frobenius norm of A. We fix the iteration of the gradient descent and assume that the weight W takes the value $W^{(0)}$ and $\widetilde{C}$ takes the value $\widetilde{C}^{(0)}$.*

We assume that the following inequality holds for all $k \in [K]$ and $l \in [K]$ uniformly $W \in \mathcal{N}_0$, where $\mathcal{N}_0$ is an open neighbourhood of $W^{(0)}$:

$$|\mathbb{E}\left[\mathrm{softmax}_l(W^\top g(x_k))\right] - \mathrm{softmax}_l(\widetilde{C}_k)| \leq \varepsilon.$$

We also assume that on $\mathcal{N}_0$, $\widetilde{C}$ can be regarded as a smooth function of W and the Frobenius norm of the Hessian is bounded on $\mathcal{N}_0$. Furthermore, we assume that the following small variance assumption with $\widetilde{\varepsilon} > 0$ for all k:

$$\sum_{m=1}^K \mathbb{V}\left[(W^\top g(x_k))_m\right] \leq \widetilde{\varepsilon}.$$

Then if $\|\Delta W\|_{\mathrm{F}}$ is sufficiently small, there exist a positive constant $c > 0$ depending only on K with $c = O(\mathrm{poly}(K))$ and a positive constant $c' > 0$ such that the following inequality holds:

$$\left| G_{ij} - \sum_{k=1}^{K} \frac{\partial \psi}{\partial \widetilde{C}_k} (\Delta W)^\top z_k \right| \leq c \left\| \frac{\partial \psi}{\partial C} \right\|_{\mathrm{F}} (\varepsilon + \widetilde{\varepsilon}) + c'(\|\Delta W\|_{\mathrm{F}}^2 + \|\Delta \widetilde{C}\|_{\mathrm{F}}^2).$$

Here $\Delta W = \tilde{\nu}_{ij}^t$ and $\widetilde{C}_k$ is a column vector such that the k-th row of $\widetilde{C}$ is given as $\widetilde{C}_k$, and we consider Jacobi matrices at $\widetilde{C} = \widetilde{C}^{(0)}$ and the corresponding value of C.

Proof. In this proof, to simplify notation, we denote $\mathrm{softmax}(z)$ by $\sigma(z)$ for $z \in \mathbb{R}^K$. In this proof, we fix the iteration of the gradient descent and assume that the weight W takes the value $W^{(0)}$ and $\widetilde{C}$ takes the value $\widetilde{C}^{(0)}$. We assume in an open neighborhood of $W^{(0)}$, we have a smooth correspondence $W \mapsto \widetilde{C}$ and that if the value of W changes from W_0 to $W_0 + \Delta W$, then $\widetilde{C}$ changes from $\widetilde{C}_0$ to $\widetilde{C}_0 + \Delta \widetilde{C}$. To prove the theorem, we introduce the following three lemmas. We note that by the assumption of the theorem and Lemma G.3, the assumption (6) of Lemma G.4 can be satisfied with

$$\varepsilon_1 = c''(\varepsilon + \widetilde{\varepsilon}),$$

where $c'' > 0$ is a constant depending only on K with $c'' = O(\mathrm{poly}(K))$. Then by Lemma G.4, there exist constants $c_1 = c_1(K)$ and $c_2 = c_2(K)$ depending on only K with $c_1, c_2 = O(\mathrm{poly}(K))$ such that the following inequality holds for all k:

$$\left\| \left. \frac{\partial C}{\partial \widetilde{C}_k} \right|_{\widetilde{C}_k = \widetilde{C}_k^{(0)}} \left(\Delta \widetilde{C}_k - (\Delta W)^\top z_k \right) \right\|_{\mathrm{F}} \leq c_1 \varepsilon_1 + c_2 (\|\Delta \widetilde{C}_k\|_{\mathrm{F}}^2 + \|(\Delta W)^\top z_k\|_{\mathrm{F}}^2). \tag{4}$$

Then, we have the following:

$$\begin{aligned}
\left| \frac{\partial \psi}{\partial \widetilde{C}} \Delta \widetilde{C} - \sum_{k=1}^{K} \frac{\partial \psi}{\partial \widetilde{C}_k} (\Delta W)^\top z_k \right| &= \left| \frac{\partial \psi}{\partial C} \frac{\partial C}{\partial \widetilde{C}} \Delta \widetilde{C} - \sum_{k=1}^{K} \frac{\partial \psi}{\partial C} \frac{\partial C}{\partial \widetilde{C}_k} (\Delta W)^\top z_k \right| \\
&= \left| \frac{\partial \psi}{\partial C} \sum_{k=1}^{K} \frac{\partial C}{\partial \widetilde{C}_k} \Delta \widetilde{C}_k - \sum_{k=1}^{K} \frac{\partial \psi}{\partial C} \frac{\partial C}{\partial \widetilde{C}_k} (\Delta W)^\top z_k \right| \\
&\leq \left\| \frac{\partial \psi}{\partial C} \right\|_{\mathrm{F}} \left\| \sum_{k=1}^{K} \frac{\partial C}{\partial \widetilde{C}_k} \left(\Delta \widetilde{C}_k - (\Delta W)^\top z_k \right) \right\|_{\mathrm{F}}
\end{aligned}$$

Here, by fixing an order on $[K] \times [K]$, we regard $\frac{\partial \psi}{\partial \widetilde{C}}$, $\frac{\partial C}{\partial \widetilde{C}}$ and $\Delta \widetilde{C}$ as a K^2-dimensional row vector, a $K^2 \times K^2$-matrix, and a K^2-dimensional column vector, respectively. Moreover, we consider Jacobi matrices at $\widetilde{C} = \widetilde{C}^{(0)}$. Then, the assertion of the theorem from this inequality, (4), Lemma G.2. □

Lemma G.2. *Under assumptions and notations in the proof of Theorem G.1, there exists a constant $c > 0$ such that*

$$\left| G_{ij} - \left. \frac{\partial \psi}{\partial \widetilde{C}} \right|_{\widetilde{C} = \widetilde{C}^{(0)}} \Delta \widetilde{C} \right| \leq c \|\Delta W\|_{\mathrm{F}}^2.$$

Proof. By the assumption of the mapping $W \mapsto \widetilde{C}$ and the Taylor's theorem, there exists $c_1 > 0$ such that

$$\left\| \Delta \widetilde{C} - \left(\frac{\partial \widetilde{C}}{\partial W} \right) \Big|_{W = W_0} \Delta W \right\|_{\mathrm{F}} \leq c_1 \|\Delta W\|_{\mathrm{F}}^2. \tag{5}$$

By definition of G_{ij}, we have the following:

$$\begin{aligned}
\left| G_{ij} - \left. \frac{\partial \psi}{\partial \widetilde{C}} \right|_{\widetilde{C} = \widetilde{C}^{(0)}} \Delta \widetilde{C} \right| &= \left| \frac{\partial \psi}{\partial W} \Delta W - \frac{\partial \psi}{\partial \widetilde{C}} \Delta \widetilde{C} \right| \\
&= \left| \frac{\partial \psi}{\partial \widetilde{C}} \frac{\partial \widetilde{C}}{\partial W} \Delta W - \frac{\partial \psi}{\partial \widetilde{C}} \Delta \widetilde{C} \right| \\
&\leq c_1 \left\| \frac{\partial \psi}{\partial \widetilde{C}} \right\|_{\mathrm{F}} \|\Delta W\|_{\mathrm{F}}^2 .
\end{aligned}$$

Here we consider Jacobi matrices at $W = W_0$ and corresponding values. The last inequality follows from the fact that the matrix norm $\|\cdot\|_{\mathrm{F}}$ is sub-multiplicative and Eq. (5). □

Lemma G.3. *Under assumptions and notations in the proof of Theorem G.1, there exist a positive constant $c = c(K)$ depending only on K with $c = O(\mathrm{poly}(K))$ such that:*

$$|\mathbb{E}\left[\sigma_l(W^\top g(x_k))\right] - \sigma_l(W^\top z_k)| \leq c \sum_{m=1}^{K} \mathbb{V}\left[(W^\top g(x_k))_m\right],$$

for any $1 \leq k, l \leq K$.

Proof. This can be proved by applying the Taylor's theorem to σ_l. We fix k, l and apply the Taylor's theorem to the function $\xi \mapsto \sigma_l(\xi)$ at $\xi = W^\top z_k = W^\top \mathbb{E}\left[g(x_k)\right]$. Then there exists $\xi_0 \in \mathbb{R}^K$ such that

$$\sigma_l(\xi) = \sigma_l(W^\top z_k) + \left.\frac{\partial \sigma_l}{\partial \xi}\right|_{\xi = W^\top z_k} (\xi - W^\top z_k) + \frac{1}{2}(\xi - W^\top z_k)^\top H_k (\xi - W^\top z_k),$$

where $H_k = \left.\frac{\partial^2 \sigma_l}{\partial \xi^2}\right|_{\xi=\xi_0}$. By noting that $\frac{\partial \sigma_l}{\partial \xi_m} = \delta_{lm}\sigma_l(\xi) - \sigma_l(\xi)\sigma_m(\xi)$ (here δ_{lm} is the Kronecker's delta), it is easy to see that there exists a constant c_l' depending only on l and K such that $\|H\|_{\mathrm{F}} < c_l'$ and $c_l' = O(\mathrm{poly}(K))$. By letting $\xi = W^\top g(x_k)$ in the above equation and taking the expectation of the both sides, we obtain the assertion of the lemma with $c = \frac{1}{2}\max_{l \leq [K]} c_l'$. □

Lemma G.4. *Under assumptions and notations in the proof of Theorem G.1, we assume there exists $\varepsilon_1 > 0$ such that the following inequality holds for all k and l for any W in an open neighborhood of $W^{(0)}$ and corresponding $\widetilde{C}$:*

$$\left|\sigma_l(W^\top z_k) - \sigma_l(\widetilde{C}_k)\right| \leq \varepsilon_1. \tag{6}$$

Furthermore, we assume that $\|(\Delta W)^\top z_k\|_{\mathrm{F}}$ is sufficiently small for all k. Then there exist constants $c_1 = c_1(K)$ and $c_2 = c_2(K)$ depending on only K with $c_1, c_2 = O(\mathrm{poly}(K))$ such that

$$\left\| \left.\frac{\partial C}{\partial \widetilde{C}_k}\right|_{\widetilde{C}_k = \widetilde{C}_k^{(0)}} \left(\Delta \widetilde{C}_k - (\Delta W)^\top z_k\right) \right\|_{\mathrm{F}} \leq c_1\varepsilon_1 + c_2(\|\Delta \widetilde{C}_k\|_{\mathrm{F}}^2 + \|(\Delta W)^\top z_k\|_{\mathrm{F}}^2).$$

Here, $\widetilde{C}_k$ (resp. $\Delta\widetilde{C}_k$) is a column vector such that the k-th row vector of $\widetilde{C}$ (resp. $\Delta\widetilde{C}$) is given as $\widetilde{C}_k$ (resp. $\Delta\widetilde{C}_k$). Moreover, when defining Jacobi matrices, we regard C as a K^2-vector and consider a $K^2 \times K$ Jacobi matrix $\left.\frac{\partial C}{\partial \widetilde{C}_k}\right|_{\widetilde{C}_k = \widetilde{C}_k^{(0)}}$ at $\widetilde{C}_k = \widetilde{C}_k^{(0)}$.

Proof. Since (6) holds all W in an open neighborhood of $W^{(0)}$ and corresponding $\widetilde{C}$, we apply the Taylor's theorem to the function $\xi \mapsto \sigma_l(\xi)$ at $\xi = (W^{(0)})^\top z_k$ and $\xi = \widetilde{C}_k^{(0)}$. Then by (6) and the same argument in the proof of Lemma G.3, we have

$$\left| \left.\frac{\partial \sigma_l}{\partial \xi}\right|_{\xi=\mu_k} \Delta\mu_k - \left.\frac{\partial \sigma_l}{\partial_\xi}\right|_{\xi=\widetilde{C}_k^{(0)}} \Delta\widetilde{C}_k \right| \leq \varepsilon_1 + c_2'(\|\Delta\mu_k\|_{\mathrm{F}}^2 + \|\Delta\widetilde{C}_k\|_{\mathrm{F}}^2),$$

where $\mu_k = (W^{(0)})^\top z_k$, $\Delta\mu_k = (\Delta W)^\top z_k$. Noting that $(\frac{\partial \sigma_l}{\partial \xi})_m$ is given as $\delta_{ml}\sigma_l(\xi) - \sigma_m(\xi)\sigma_l(\xi)$, (6) and the assumption that $\|\mu_k\|_{\mathrm{F}}$ is sufficiently small, we see that there exists a constant $c_1', c_2' > 0$ depending only on K with $c_1', c_2' = O(\mathrm{poly}(K))$ such that the following inequality holds:

$$\left| \left.\frac{\partial \sigma_l}{\partial \xi}\right|_{\xi=\widetilde{C}_k^{(0)}} \left(\Delta\mu_k - \Delta\widetilde{C}_k\right) \right| \leq c_1'\varepsilon_1 + c_2'(\|\Delta\mu_k\|_{\mathrm{F}}^2 + \|\Delta\widetilde{C}_k\|_{\mathrm{F}}^2). \tag{7}$$

Next, we consider entries of the K^2-vector $\left.\frac{\partial C}{\partial \widetilde{C}_k}\right|_{\widetilde{C}_k = \widetilde{C}_k^{(0)}} (\Delta\widetilde{C}_k - \Delta\mu_k)$. Here as previously mentioned by fixing an order on $[K] \times [K]$, we regard $\frac{\partial C}{\partial \widetilde{C}_k}$ as a $K^2 \times K$-matrix. For $(k, l) \in [K] \times [K]$, by the definition of the mapping $\widetilde{C} \mapsto C$,

(k,l)-th entry of $\frac{\partial C}{\partial \widetilde{C}_k}\Big|_{\widetilde{C}_k=\widetilde{C}_k^{(0)}}(\Delta\widetilde{C}_k - \Delta\mu_k)$ is given as $\pi_k \frac{\partial \sigma_l}{\partial \xi}\Big|_{\xi=\widetilde{C}_k^{(0)}}(\Delta\widetilde{C}_k - \Delta\mu_k)$. By (7), we see that there exist constants c_1'', c_2'' depending only on K and $c_1'', c_2'' = O(\mathrm{poly}(K))$ such that

$$\left\| \frac{\partial C}{\partial \widetilde{C}_k}\Big|_{\widetilde{C}_k=\widetilde{C}_k^{(0)}} \left(\Delta\widetilde{C}_k - (\Delta W)^\top z_k\right) \right\|_{\mathrm{F}} \leq c_1''\varepsilon_1 + c_2''(\|\Delta\widetilde{C}_k\|_{\mathrm{F}}^2 + \|(\Delta W)^\top z_k\|_{\mathrm{F}}^2).$$

Since constants c_1'', c_2'' may depend on (k,l) by taking $c_1 = \max_{(k,l)} c_1''$ and $c_2 = \max_{(i,l)} c_2''$, we have the assertion of the lemma. □

G.2. Proof of Theorem F.1

First we introduce the following lemma, which is (essentially) due to (Freund & Schapire, 1997). Although, one can prove the following result by a standard argument, since our problem setting is different, we provide a proof for the sake of completeness.

Lemma G.5 (c.f. (Freund & Schapire, 1997)). *We assume that $G_{i,j}^{(t)} \in [0,1]$ for all t and $1 \leq i,j \leq K$. For $(i,j) \in [K]\times[K]$, we define $\overline{S}_{i,j} = \sum_{t=1}^T \mathbb{E}\left[G_{i,j}^{(t)}\right]$. For a policy $\boldsymbol{\mathcal{P}} = (\mathcal{P}_t)_{t=1}^T$, we define $\overline{S}_{\boldsymbol{\mathcal{P}}} := \sum_{t=1}^T \mathbb{E}\left[G_{i_t,j_t}^{(t)}\right]$. Then, we have the following inequality:*

$$-2\log K + s \max_{(i,j)\in[K]\times[K]} \overline{S}_{i,j} \leq (\exp(s)-1)\overline{S}_{\boldsymbol{\mathcal{P}}_{\mathrm{SelMix}}}.$$

Proof. This lemma can be proved by a standard argument, but for the sake of completeness, we provide a proof. We put $\mathcal{A} = [K]\times[K]$, $a_t = (i^{(t)}, j^{(t)})$ and in the proof we simply denote $\boldsymbol{\mathcal{P}}_{\mathrm{SelMix}}$ by $\boldsymbol{\mathcal{P}}$. For $a \in \mathcal{A}$ and $1 \leq t \leq T+1$, we define $w_{a,t}$ as follows. We define $w_{a,1} = 1/K^2$ for all $a \in \mathcal{A}$ and $w_{a,t+1} = w_{a,t}\exp(sG_a^{(t)})$. We also define $W_t = \sum_{a\in\mathcal{A}} w_{a,t}$. Then, the distribution $\mathcal{P}_t$ is given as the probability $(w_{a,t}/W_t)_{a\in\mathcal{A}}$ by definition. Noting that $\exp(sx) \leq 1 + (\exp(s)-1)x$ for $x \in [0,1]$, we have the following inequality:

$$\begin{aligned} W_{t+1} &= \sum_{a\in\mathcal{A}} w_{a,t+1} = \sum_{a\in\mathcal{A}} w_{a,t}\exp(sG_a^{(t)}) \\ &\leq \sum_{a\in\mathcal{A}} w_{a,t}(1+\exp(s-1)G_a^{(t)}). \end{aligned}$$

Thus, we have

$$\begin{aligned} W_{t+1} &\leq \sum_{a\in\mathcal{A}} w_{a,t}(1+\exp(s-1)G_a^{(t)}) \\ &= W_t\left(1 + (\exp(s)-1)\mathbb{E}_{\mathcal{P}_t}\left[G_{a_t}^{(t)}\right]\right), \end{aligned}$$

where $\mathbb{E}_{\mathcal{P}_t}[\cdot]$ denotes the expectation with respect to a_t. By repeatedly apply the inequality above, we obtain:

$$W_{T+1} \leq \prod_{t=1}^T \left(1 + (\exp(s)-1)\mathbb{E}_{\mathcal{P}_t}\left[G_{a_t}^{(t)}\right]\right).$$

Let $a \in \mathcal{A}$ be any pair. By this inequality and $W_{T+1} \geq w_{a,T+1} = \frac{1}{K^2}\exp(s\sum_{t=1}^T G_a^{(t)})$, we have the following:

$$\frac{1}{K^2}\exp(s\sum_{t=1}^T G_a^{(t)}) \leq \prod_{t=1}^T \left(1 + (\exp(s)-1)\mathbb{E}_{\mathcal{P}_t}\left[G_{a_t}^{(t)}\right]\right).$$

By taking $\log$ of both sides and $\log(1+x) \leq x$, we have

$$\begin{aligned} -2\log K + s\sum_{t=1}^T G_a^{(t)} &\leq \sum_{t=1}^T \log\left(1 + (\exp(s)-1)\mathbb{E}_{\mathcal{P}_t}\left[G_{a_t}^{(t)}\right]\right) \\ &\leq (\exp(s)-1)\sum_{t=1}^T \mathbb{E}_{\mathcal{P}_t}\left[G_{a_t}^{(t)}\right]. \end{aligned}$$

By taking the expectation with respect to the randomness of $G_{i,j}^{(t)}$, we obtain the following:

$$-2\log K + s\overline{S}_a \leq (\exp(s)-1)\overline{S}_{\boldsymbol{\mathcal{P}}}.$$

Since $a \in [K]\times[K]$ is arbitrary, we have the assertion of the lemma. □

We can prove Theorem F.1 by Lemma G.5 as follows:

Proof of Theorem F.1. Let (i^*, j^*) be the best fixed Mixup pair hindsight, i.e., $(i^*, j^*) = \mathrm{argmax}_{(i,j)\in[K]\times[K]} \overline{S}_{i,j}$. Since any non-adaptive (or stationary) policy is no better than $\delta_{(i^*,j^*)}$, to prove the theorem, it is enough to prove the following:

$$\overline{S}_{i^*,j^*} \leq \overline{S}_{\boldsymbol{\mathcal{P}}} + 2\sqrt{T\log K}. \tag{8}$$

Here in this proof, we simply denote $\boldsymbol{\mathcal{P}}_{\mathrm{SelMix}}$ by $\boldsymbol{\mathcal{P}}$. To prove (8), we define the pseudo regret R_T by $R_T = \overline{S}_{i^*,j^*} - \overline{S}_{\boldsymbol{\mathcal{P}}}$. Then by Lemma G.5, we have

$$R_T \leq \frac{(\exp(s)-1-s)\overline{S}_{i^*,j^*} + 2\log K}{\exp(s)-1}.$$

We put $s = \log(1+\alpha)$ with $\alpha > 0$. Then we have

$$R_T \leq \frac{(\alpha - \log(1+\alpha))\overline{S}_{i^*,j^*} + 2\log K}{\alpha}.$$

We note that the following inequality holds for $\alpha > 0$:

$$\frac{\alpha - \log(1+\alpha)}{\alpha} \leq \frac{1}{2}\alpha$$

Then it follows that:

$$R_T \leq \frac{1}{2}\alpha\overline{S}_{i^*,j^*} + \frac{2\log K}{\alpha} \leq \frac{1}{2}\alpha T + \frac{2\log K}{\alpha}.$$

Here the second inequality follows from $\overline{S}_{i^*,j^*} \leq T$. We take $\alpha = 2\sqrt{\frac{\log K}{T}}$. Then we have $R_T \leq 2\sqrt{T\log K}$. Thus, we have the assertion of the theorem. □

G.3. Proof of Theorem F.2

Proof. This can be proved by standard argument of the proof of the mirror descent method (see e.g. (Lattimore & Szepesvári, 2020), chapter 28).

Denote by $\Delta \subset \mathbb{R}^{K\times K}$ the probability simplex of dimension K^2-1. Let $(i_0, j_0) \in K\times K$ be the best fixed mixup hindsight. Since any non-adaptive policy is no better than the best fixed mixup in terms of $\overline{G}$, we may assume that $\boldsymbol{\mathcal{P}}^{(0)} = (\pi_0)_t$, where π_0 is the one-hot vector in Δ defined as $(\pi_0)_{ij} = 1$ if $(i,j) = (i_0, j_0)$ and 0 otherwise for $1 \leq i, j \leq K$. Let F be the negative entropy function, i.e., $F(p) = \sum_{i,j=1}^K p_{ij}\log p_{ij}$. For $p \in \Delta$ and $G \in \mathbb{R}^{K\times K}$, we define $\langle p, G\rangle = \sum_{i,j=1}^K p_{ij}G_{ij}$. Then, it is easy to see that $p^{(t)} = \mathcal{P}_{\mathrm{SelMix}}^{(t)}$ defined above is given as the solution of the following:

$$p^{(t)} = \mathrm{argmin}_{p\in\Delta} -s\langle p, G^{(t)}\rangle + D(p, p^{(t-1)}). \tag{9}$$

Here D denotes the KL-divergence and we define $p^{(0)} = (1/K^2)_{1\leq i,j\leq K} = \mathrm{argmin}_{p\in\Delta} F(p)$. Since the optimization problem (9) is a convex optimization problem, by the first order optimality condition, we have

$$\langle \pi_0 - p^{(t)}, G^{(t)}\rangle \leq \frac{1}{s}\left\{D(\pi_0, p^{(t-1)}) - D(\pi_0, p^{(t)}) - D(p^{(t)}, p^{(t-1)})\right\}.$$

By summing the both sides and taking expectation, we have

$$\begin{aligned} T\overline{G}^{(T)}(\boldsymbol{\mathcal{P}}^{(0)}) - T\overline{G}^{(T)}(\boldsymbol{\mathcal{P}}_{\mathrm{SelMix}}) &\leq \frac{1}{s}\left\{D(\pi_0, p^{(0)}) - D(\pi_0, p^{(T)}) - \sum_{t=1}^T D(p^{(t)}, p^{(t-1)})\right\} \\ &\leq \frac{1}{s}D(\pi_0, p^{(0)}). \end{aligned}$$

Here the second inequality follows from the non-negativity of the KL-divergence. Since $p^{(0)} = \operatorname{argmin}_p F(p)$, by the first-order optimality condition, we have $D(\pi_0, p^{(0)}) \leq F(\pi_0) - F(p^{(0)})$. Noting that $F(\pi_0) \leq 0$, we have the following

$$T\overline{G}^{(T)}(\boldsymbol{\mathcal{P}}^{(0)}) - T\overline{G}^{(T)}(\boldsymbol{\mathcal{P}}_{\text{SelMix}}) \leq \frac{-F(p^{(0)})}{s} = \frac{\log K^2}{s}.$$

This completes the proof. □

H. Unconstrained Derivatives of metric

For any general metric $\psi(C[h])$ the derivative w.r.t the uncostrained confusion matrix $\tilde{C}[h]$ is expressible purely in terms of the entries of the confusion matrix. The derivative using chain rule is expressed as follows,

$$\frac{\partial \psi(C[h])}{\partial \tilde{C}_{ij}[h]} = \sum_{k,l} \frac{\partial \psi(C[h])}{\partial C_{kl}[h]} \cdot \frac{\partial C_{kl}[h]}{\partial \tilde{C}_{ij}[h]} \tag{10}$$

We observe that in Eq. 10 the partial derivative $\frac{\partial \psi(C[h])}{\partial C_{kl}[h]}$ is purely a function of entries of $C[h]$ since $\psi(C[h])$ itself is a function of entries of $C[h]$. The second term is the partial derivative of our confusion matrix w.r.t the unconstrained confusion matrix. Since C and $\tilde{C}$ are related by the following relation $C_{ij}[h] = \text{softmax}(\tilde{C}_i[h])_j$. By virtue of the aforementioned map $\frac{\partial C_{kl}[h]}{\partial \tilde{C}_{ij}[h]}$ also happens to be expressible in terms of $C[h]$:

$$\frac{\partial C_{kl}[h]}{\partial \tilde{C}_{ij}[h]} = \begin{cases} 0, & k \neq i \\ -\frac{C_{il}[h] \cdot C_{ij}[h]}{\pi_i^{\text{val}}}, & i = k, l \neq j \\ C_{ij}[h] - \frac{C_{ij}^2[h]}{(\pi_i^{\text{val}})^2}, & i = k, l = j \end{cases} \tag{11}$$

Let us consider the metric mean recall $\psi^{\text{AM}}(C[h]) = \frac{1}{K} \sum_i \frac{C_{ii}[h]}{\sum_j C_{ij}[h]}$. The derivative of $\psi^{\text{AM}}(C[h])$ w.r.t the unconstrained confusion matrix $\tilde{C}$ can be expressed in terms of the entries of the confusion matrix. This is a useful property of this partial derivative since we need not infer the inverse map from $C \to \tilde{C}$ inorder to evaluate the partial derivative in terms of $\tilde{C}$. It can be expressed follows:

$$\frac{\partial \psi^{\text{AM}}(C[h])}{\partial \tilde{C}_{ij}[h]} = \begin{cases} -\frac{C_{ij}[h] \cdot C_{ii}[h]}{K(\pi_i^{\text{val}})^2}, & m \neq n \\ \frac{C_{ii}[h]}{K \cdot \pi_i^{\text{val}}} - \frac{C_{ii}^2[h]}{K \cdot (\pi_i^{\text{val}})^2}, & m = n \end{cases} \tag{12}$$

Hence we can conclude that for a metric defined as a function of the entries of the confusion matrix, the derivative w.r.t the unconstrained confusion matrix ($\tilde{C}$) is easily expressible using the entries of the confusion matrix (C).

I. Updating the Lagrange multipliers

I.1. Min. Recall and Min of Head and Tail Recall

Consider the objective $\psi^{\text{MR}}(C[h]) = \sum \min_{\boldsymbol{\lambda} \in \Delta_{K-1}} \sum_{i \in [K]} \lambda_i \text{Rec}_i[h] = \sum \min_{\boldsymbol{\lambda} \in \Delta_{K-1}} \sum_{i \in [K]} \lambda_i \frac{C_{ii}[h]}{\sum_{j \in [K]} C_{ij}[h]}$, as in Table D.1. the lagrange multipliers are sampled from a $K-1$ dimensional simplex and $\lambda_i = 1$ if recall of i^{th} class is the lowest and the remaining lagrange multiplers are zero. Hence, a good approximation of the lagrange multipliers at a given time step t can be expressed as:

$$\lambda_i^t = \frac{e^{-\omega \text{Rec}_i[h]}}{\sum_{j \in [K]} e^{-\omega \text{Rec}_j[h]}} \tag{13}$$

This has some nice properties such as the Lagrange multipliers being a soft and momentum free approximation of their hard counter part. For sufficiently high ω this approximates the objective to the min recall.

I.2. Mean recall under Coverage constraints

For the objective $\psi^{\text{AM}}_{\text{cons.}}(C[h]) = \min_{\boldsymbol{\lambda}\in\mathbb{R}^K_+} \sum_{i\in[K]} \text{Rec}_i[h] + \sum_{j\in[K]} \lambda_j \left(\text{Cov}_j[h] - \frac{\alpha}{K}\right) = \min_{\boldsymbol{\lambda}\in\mathbb{R}^K_+} \sum_{i\in[K]} \frac{C_{ii}[h]}{\sum_{j\in[K]} C_{ij}[h]} + \sum_{j\in[K]} \lambda_j \left(\sum_{i\in[K]} C_{ij}[h] - \frac{\alpha}{K}\right)$. For practical purposes, we look at a related constrained optimization problem,

$$\psi^{\text{AM}}_{\text{cons.}}(C[h]) = \min_{\boldsymbol{\lambda}\in\mathbb{R}^K_+} \frac{1}{\lambda_{max}+1} \left(\sum_{i\in[K]} \text{Rec}_i[h] + \sum_{j\in[K]} \lambda_j \left(\text{Cov}_j[h] - \frac{\alpha}{K}\right)\right)$$

Such that if $\left(\text{Cov}_i[h] - \frac{\alpha}{K}\right) < 0$, then λ_i increases, and vice-versa for the converse case. Also, if $\exists i$ s.t. $\left(\text{Cov}_i[h] - \frac{\alpha}{K}\right) < 0$, then this implies that $\frac{1}{\lambda_{\text{max}}+1} \to 0^+$ and $\frac{\lambda_{\text{max}}}{\lambda_{\text{max}}+1} \to 1^-$, which forces h to satisfy the constraint $\left(\text{Cov}_i[h] - \frac{\alpha}{K}\right) > 0$. Based on this, a momentum free formulation for updating the lagrangian multipliers is as follows:

$$\lambda_i = \max\left(0, \Lambda_{\text{max}}\left(1 - e^{\frac{\text{Cov}_i[h] - \frac{\alpha}{K}}{\tau}}\right)\right)$$

Here, λ_{max} is the maximum value that the Lagrange multiplier can take. A large value of λ_{max} forces the model to focus more on the coverage constraints that to be biased towards mean recall optimization. τ is a hyperparameter that is usually kept small, say 0.01 or so, which acts as sort of a tolerance factor to keep the constraint violation in check.

J. Experimental details

J.1. Hyperparameter Table

The detailed values of all hyperparameters specific to each dataset has been mentioned in Table J.1 and Table J.2.

Table J.1. Table depicting Hyperparameters across our experiments for the supervised classification task.

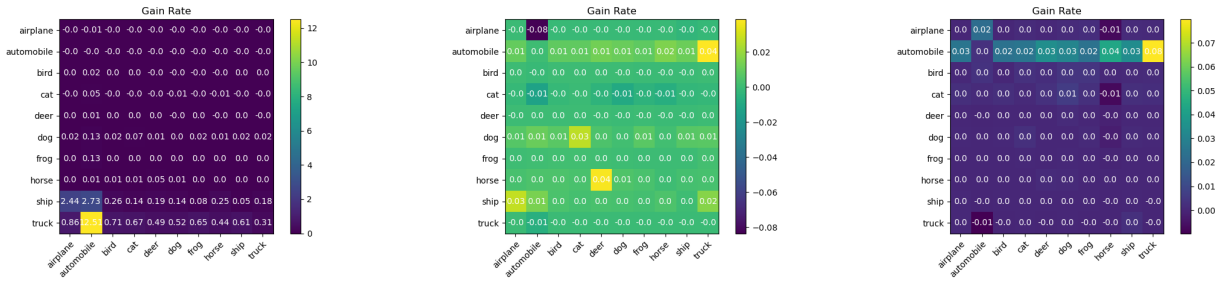
Parameter	CIFAR-10 ($\rho = 100$)	CIFAR-100 ($\rho = 100$)	Imagenet-1k LT
Gain scaling (s)	10.0	10.0	10.0
$\omega_{\text{Min. Rec}}$	50	25	100
λ_{max}	100	100	100
τ	0.01	0.01	0.001
α	0.95	0.95	0.95
Batch Size	128	128	256
Learning Rate(f)	3e-3	3e-3	0.1
Learning Rate(g)	3e-4	3e-4	0.01
Optimizer	SGD	SGD	SGD
Scheduler	Cosine	Cosine	Cosine
Total SGD Steps	2k	2k	2.5k
Resolution	32 X 32	32 X 32	224 X 224
Arch.	ResNet-32	ResNet-32	ResNet-50

J.2. Computational Requirements

The experiments were done on Nvidia A5000 GPU(24 GB). While the fine-tuning was done on a single A5000, the pre-training was done using Pytorch data parallel on 4XA5000. The pre-training was done until no significant change in metrics was observed and the fine-tuning was done for 10k steps of SGD with a validation step every 50 steps. A major advantage of SelMix over CSST is that the process of training a model optimized for a specific objective requires end to end training which is computationally expensive($\sim$10 hrs on CIFAR datasets). Our finetuning method takes a fraction ($\sim$1hr) of what it requires in computing time compared to CSST.

Table J.2. Table of Hyperparameters for Semi-Supervised datasets.

Parameter	CIFAR-10 (All distributions)	CIFAR-100 ($\rho_l = 10, \rho_u = 10$)	STL-10	Imagenet-100($\rho_l = \rho_u = 10$)
Gain scaling (s)	10.0	10.0	2.0	10.0
$\omega_{\text{Min. Rec}}$	40	20	20	20
$\lambda_{\max}$	100	100	100	100
τ	0.01	0.01	0.01	0.01
α	0.95	0.95	0.95	0.95
Batch Size	64	64	64	64
Learning Rate(f)	3e-4	3e-4	3e-4	0.1
Learning Rate(g)	3e-5	3e-5	3e-5	0.01
Optimizer	SGD	SGD	SGD	SGD
Scheduler	Cosine	Cosine	Cosine	Cosine
Total SGD Steps	10k	10k	10k	10k
Resolution	32 X 32	32 X 32	32 X 32	224 X 224
Arch.	WRN-28-2	WRN-28-2	WRN-28-2	WRN-28-2



(a) Initial Stage ($t = 0$ SGD steps)

(b) Intermediate Stage ($t = 5k$ SGD steps)

(c) Final Stage ($t = 10k$ SGD steps)

Figure L.1. Evolution of gain matrix for mean recall optimized run for CIFAR-10 LT ($\rho_l = \rho_u$)

K. Optimization of H-mean with Coverage Constraints for Semi-Supervised Learning

We consider the objective of optimizing H-mean subject to the constraint that all classes must have a coverage $\geq \frac{\alpha}{K}$. For CIFAR-10, when the unlabeled data distribution matches the labeled data distribution, uniform or inverted, SelMix is able to satisfy the coverage constraints. A similar observation could be made for CIFAR-100, where the constraint is to have the minimum head and tail class coverage above $\frac{0.95}{K}$. For STL-10, SelMix fails to satisfy the constraint because the validation dataset is minimal (500 samples compared to 5000 in CIFAR). We want to convey here that as CSST is only able to optimize for linear metrics like min. recall its performance is inferior on complex objectives like optimizing H-mean with constraints. This shows the superiority of the proposed SelMix framework.

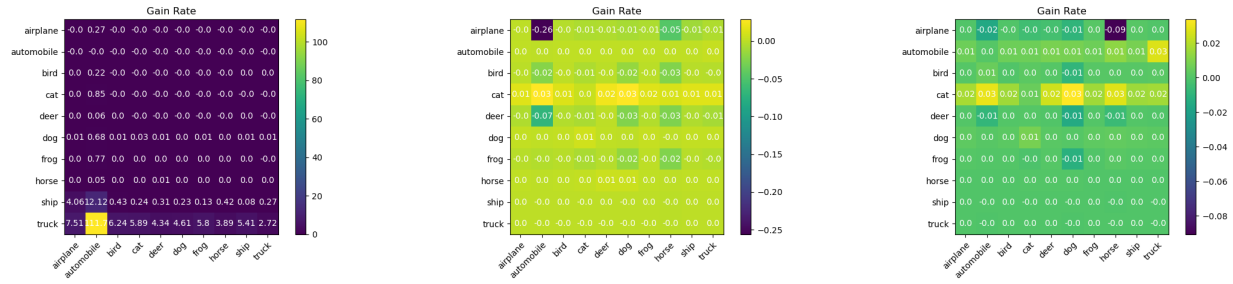
Table J.3. The Expression of Non-Decomposable Objectives we consider in our paper.

Metric	Definition
Mean Recall (ψ^{AM})	$\frac{1}{K}\sum_{i\in[K]}\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}$
G-mean (ψ^{GM})	$\left(\prod_{i\in[K]}\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}\right)^{\frac{1}{k}}$
H-mean (ψ^{HM})	$K\left(\sum_{i\in[K]}\frac{\sum_{j\in[K]}C_{ij}[h]}{C_{ii}[h]}\right)^{-1}$
Min. Recall (ψ^{MR})	$\min_{\boldsymbol{\lambda}\in\Delta_{K-1}}\sum_{i\in[K]}\lambda_i\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}$
Min of Head and Tail class recall (ψ^{MR}_{HT})	$\min_{(\lambda_{\mathcal{H}},\lambda_{\mathcal{T}})\in\Delta_1}\frac{\lambda_{\mathcal{H}}}{\lvert\mathcal{H}\rvert}\sum_{i\in\mathcal{H}}\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}$ $+\frac{\lambda_{\mathcal{T}}}{\lvert\mathcal{T}\rvert}\sum_{i\in\mathcal{T}}\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}$
Mean Recall s.t. per class coverage $\geq\frac{\alpha}{K}$ ($\psi^{AM}_{\text{cons.}}$)	$\min_{\boldsymbol{\lambda}\in\mathbb{R}^K_+}\frac{1}{K}\sum_{i\in[K]}\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}+\sum_{j\in[K]}\lambda_j\left(\sum_{i\in[K]}C_{ij}[h]-\frac{\alpha}{K}\right)$
Mean Recall s.t. minimum of head and tail class coverage $\geq\frac{\alpha}{K}$ ($\psi^{AM}_{\text{cons.(HT)}}$)	$\min_{(\lambda_{\mathcal{H}},\lambda_{\mathcal{T}})\in\mathbb{R}^2_{\geq 0}}\frac{1}{K}\sum_{i\in[K]}\frac{C_{ii}[h]}{\sum_{j\in[K]}C_{ij}[h]}+\lambda_{\mathcal{H}}\left(\sum_{i\in[K],j\in\mathcal{H}}\frac{C_{ij}[h]}{\lvert\mathcal{H}\rvert}-\frac{0.95}{K}\right)$ $+\lambda_{\mathcal{T}}\left(\sum_{i\in[K],j\in\mathcal{T}}\frac{C_{ij}[h]}{\lvert\mathcal{T}\rvert}-\frac{0.95}{K}\right)$
H-mean s.t. per class coverage $\geq\frac{\alpha}{K}$ ($\psi^{HM}_{\text{cons.}}$)	$\min_{\boldsymbol{\lambda}\in\mathbb{R}^K_+}K\left(\sum_{i\in[K]}\frac{\sum_{j\in[K]}C_{ij}[h]}{C_{ii}[h]}\right)^{-1}+\sum_{j\in[K]}\lambda_j\left(\sum_{i\in[K]}C_{ij}[h]-\frac{\alpha}{K}\right)$
H-mean s.t. minimum of head and tail class coverage $\geq\frac{\alpha}{K}$ ($\psi^{HM}_{\text{cons.(HT)}}$)	$\min_{(\lambda_{\mathcal{H}},\lambda_{\mathcal{T}})\in\mathbb{R}^2_{\geq 0}}K\left(\sum_{i\in[K]}\frac{\sum_{j\in[K]}C_{ij}[h]}{C_{ii}[h]}\right)^{-1}+\lambda_{\mathcal{H}}\left(\sum_{i\in[K],j\in\mathcal{H}}\frac{C_{ij}[h]}{\lvert\mathcal{H}\rvert}-\frac{0.95}{K}\right)$ $+\lambda_{\mathcal{T}}\left(\sum_{i\in[K],j\in\mathcal{T}}\frac{C_{ij}[h]}{\lvert\mathcal{T}\rvert}-\frac{0.95}{K}\right)$

Table K.1. Comparison of methods for optimization of H-mean with coverage constraints.

	CIFAR-10 $\rho_l=100,\rho_u=\frac{1}{100}$ $N_1=1500,M_1=30$		**CIFAR-10** $\rho_l=\rho_u=100$ $N_1=1500,M_1=3000$		**CIFAR-10** $\rho_l=100,\rho_u=1$ $N_1=1500,M_1=3000$		**CIFAR-100** $\rho_l=\rho_u=10$ $N_1=150,M_1=300$		**STL-10** $\rho_l=10,\rho_u=$ NA $N_1=450,\sum_i M_i=100$k	
	HM	Min Cov.	HM	Min Cov.	HM	Min Cov.	HM	Min H-T Cov.	HM	Min Cov.
DARP	78.1±0.9	0.065±3e-3	81.9±0.5	0.070±3e-3	83.5±0.8	0.067±3e-3	48.7±1.3	0.0040±2e-3	74.0±0.5	0.058±2e-3
CReST	65.8±1.5	0.040±5e-3	81.0±0.7	0.073±5e-3	84.6±0.2	0.075±7e-4	48.3±0.2	0.0083±2e-4	67.1±1.1	0.066±2e-3
DASO	78.1±0.1	0.072±3e-3	83.5±0.3	0.083±1e-3	88.4±0.5	0.089±1e-3	49.1±0.7	0.0063±3e-4	76.6±1.1	<u>0.083</u>±3e-3
ABC	<u>79.6</u>±0.3	0.073±5e-3	<u>84.6</u>±0.5	0.086±3e-3	88.2±0.7	0.086±1e-3	<u>50.1</u>±1.2	0.0089±2e-4	74.7±1.5	0.079±7e-3
CSST	76.5±4.9	<u>0.081</u>±6e-3	76.9±0.2	<u>0.093</u>±3e-4	86.7±0.7	<u>0.092</u>±1e-3	47.7±0.8	<u>0.0098</u>±2e-4	<u>78.3</u>±2.6	0.081±6e-3
FixMatch (LA)	78.3±0.8	0.064±1e-3	76.7±0.1	0.056±3e-3	<u>89.3</u>±0.2	0.086±1e-3	45.5±2.1	0.0053±1e-4	74.6±1.7	0.066±5e-3
w/SelMix (Ours)	**81.0**±0.8	**0.095**±1e-3	**85.1**±0.1	**0.095**±1e-3	**91.3**±0.7	**0.096**±1e-3	**53.8**±0.5	**0.0098**±1e-4	**79.1**±1.2	**0.088**±1e-3

L. Evolution of Gain Matrix with Training



(a) Initial Stage ($t = 0$ SGD steps)

(b) Intermediate Stage ($t = 3k$ SGD steps)

(c) Final Stage ($t = 10k$ SGD steps)

Figure L.2. Evolution of gain matrix for min. recall optimized run for CIFAR-10 LT ($\rho_u = \rho_l$).

From the above collection of gain matrices, which are taken from different time steps of the training phase, we observe that the range ($|\max(\mathbf{G}^{(t)}) - \min(\mathbf{G}^{(t)})|$) of the gain matrix decreases with increase in SGD steps t, and settles on a negligible value by the time training is finished. This could be attributed to the fact that as the training progresses, the marginal improvement of the gain matrix decreases.

Another phenomenon we observe is that initially, during training, only a few mixups (particularly tail class ones) have a disproportionate amount of gain associated with them. A downstream consequence of this is that the sampling function $\mathcal{P}_{\text{SelMix}}$ prefers only a few (i, j) mixups. Whereas, as the training continues, it becomes more exploratory rather than greedily exploiting the mixups that give the maximum gain at a particular timestep.

M. Detailed Analysis of SelMix Models

In this section, we analyze all these specific models on all other sets of metrics. We tabulate our results in Table M.1. It can be observed that when the model is trained for the particular metric for the diagonal entries, it performs the best on it. Also, we generally find that all models trained through SelMix reasonably perform on other metrics. This demonstrates that the models produced are balanced and fair in general. As a rule of thumb, we would like the users to utilize models trained for constrained objectives as they perform better than others cumulatively.

Table M.1. Values of all metric values for individually optimized runs for CIFAR-10 LT Semi-Supervised ($\rho_l = \rho_u$)

Optimized On \ Observed Metric	Mean Rec.	Min. Rec.	HM	GM	Mean Rec./Min Cov.	HM/Min Cov.
Mean Rec.	85.4	77.6	85.0	85.1	85.4/0.089	85.0/0.089
Min. Rec.	84.2	79.1	84.1	84.2	84.2/0.091	84.1/0.091
HM	85.3	77.7	85.1	85.2	85.3/0.091	85.1/0.091
GM	85.3	77.5	85.1	85.3	85.3/0.091	85.1/0.091
Mean Rec./Min. Cov.	85.7	75.9	84.7	84.8	85.7/0.095	84.7/0.095
HM/Min Cov.	85.1	76.2	84.8	84.9	85.1/0.095	84.8/0.095

N. Comparison between FixMatch and FixMatch (LA)

We find that using logit-adjusted loss helps in training feature extractors, which perform much superior in comparison to the vanilla FixMatch Algorithm (Table N.1). However, our method SelMix is able to improve both the FixMatch and the FixMatch (LA) variant. We advise users to use the FixMatch (LA) algorithm for better results.

Table N.1. Comparison of the FixMatch and FixMatch (LA) methods with SelMix.

Method	Mean *Recall*	Min *Coverage*	Min *Recall*	Mean *Recall*
FixMatch	76.8	0.037	36.7	76.8
w/ SelMix	84.7	0.094	78.8	82.7
FixMatch (LA)	82.6	0.065	63.6	82.6
w/ SelMix	85.4	0.095	79.1	84.1

O. Variants of Mixup

As SelMix is a distribution on which class samples $\mathbf{i}, \mathbf{j}$ to be mixed up, it can be easily be combined with different variants of mixup (Yun et al., 2019; Kim et al., 2020b). To demonstrate this, we replace the feature mixup that we perform in SelMix, with CutMix and PuzzleMix. Table O.1 contains results for various combinations for optimizing the Mean Recall and Min Recall across cases. We observe that SelMix can optimize the desired metric, even with CutMix and PuzzleMix. However, the feature mixup we performed originally in SelMix works best in comparison to other variants. This establishes the complementarity of SelMix with the different variants of Mixup like CutMix, PuzzleMix, etc., which re-design the procedure of mixing up images.

Table O.1. Comparison of SelMix when applied to various Mixup variants.

Method	Mean *Recall*	Min *Recall*
FixMatch	$79.7_{\pm 0.6}$	$55.9_{\pm 1.9}$
w/ SelMix (CutMix)	$84.8_{\pm 0.2}$	$75.3_{\pm 0.1}$
w/ SelMix (PuzzleMix)	$85.1_{\pm 0.3}$	$75.2_{\pm 0.1}$
w/ SelMix (Features-Ours)	$85.4_{\pm 0.1}$	$79.1_{\pm 0.1}$

O.1. Extention to Semi-Supervised Learning

We show the efficacy of SelMix for optimizing a wide variety of Non-Decomposable metrics across different distributions for the labeled and unlabeled data. This includes cases where the underlying label distribution for labeled and unlabeled data are matched, mismatched, and even unknown for the latter. Following a similar convention to long-tail (LT) classification problems, we index the classes from $1...K$ where the distribution of labeled samples follows an exponentially decaying function. N_i and M_i denotes the number of samples in the i^{th} class of the labeled and unlabeled set respectively. The underlying label distribution of these samples is characterized by their imbalance factor ρ. Corresponding to this imbalance factor, the label distribution is defined using a factor γ as follows:

$$N_i = N_1 \cdot \gamma_l^{i-1}, M_i = M_1 \cdot \gamma_u^{i-1} \text{where } \gamma = \rho^{-\frac{1}{K-1}}$$

Since the labeled and unlabeled data need not have the same distribution in our experiments, their corresponding imbalance factors shall be denoted by ρ_l and ρ_u, respectively. For CIFAR-10, we set $N_1 = 1500$ with an imbalance factor $\rho_l = \frac{N_1}{N_K} = 100$. We consider three settings for the unlabeled data where the nature of the underlying label distribution is either matched ($M_1 = 3000, \rho_u = 100$), uniform ($M_1 = 3000, \rho_u = 1$) or inverted ($M_1 = 30, \rho_u = \frac{1}{100}$). For STL-10 we use $N_1 = 450$ with an imbalance $\rho_l = 10$. The underlying labels of STL-10 for the unlabeled set (100k images) is not known $\rho_u = NA, M_i = NA$.

Table O.2. Results for sampling policies for $\mathcal{P}_{\text{Mix}}$ (CIFAR-10 LT, semi-supervised) $\rho = 100$.

Method	Mean *Recall*	Min *Coverage*	Min *Recall*	Mean *Recall*
Uniform Policy	83.3	0.072	70.5	83.3
Greedy Policy	83.6	0.093	78.2	81.8
SelMix Policy	**84.9**	**0.094**	**79.1**	**84.1**

Table O.3. Comparison on metric objectives for CIFAR-10 LT under $\rho_l \neq \rho_u$ assumption. Our experiments involve $\rho_u = 100, \rho_l = 1$ (uniform) and $\rho_u = 100, \rho_l = \frac{1}{100}$ (inverted). **SelMix** achieves significant gains over other SSL-LT methods across all the metrics.

	CIFAR-10 ($\rho_l = 100, \rho_u = \frac{1}{100}, N_1 = 1500, M_1 = 30$)					**CIFAR-10** ($\rho_l = 100, \rho_u = 1, N_1 = 1500, M_1 = 3000$)				
	Mean Rec.	Min Rec.	HM	GM	Mean Rec./Min Cov.	Mean Rec.	Min Rec.	HM	GM	Mean Rec./Min Cov.
DARP	79.7±0.8	60.7±2.4	78.1±0.9	78.9±0.9	79.7±0.8/0.065±2e-3	84.8±0.7	66.9±3.1	83.5±0.8	85.2±0.7	84.8±0.7/0.067±3e-3
CReST	71.3±0.9	40.3±3	65.8±1.5	68.6±1.2	71.3±0.9/0.040±5e-3	85.7±0.3	68.7±1.7	84.6±0.14	85.1±0.1	85.7±0.3/0.075±7e-4
CReST+	72.8±0.8	45.2±2.5	68.4±1.3	70.6±1.1	72.8±0.8/0.047±3e-3	86.4±0.2	71.7±1.9	85.6±0.2	86.1±0.1	86.4±0.2/0.078±1e-3
DASO	79.2±0.2	64.5±1.8	78.1±0.1	78.6±0.8	79.2±0.2/0.072±3e-3	88.6±0.4	78.2±1.6	88.4±0.5	88.5±0.4	88.6±0.4/0.089±1e-3
ABC	80.8±0.4	65.1±0.8	79.6±0.3	80.7±0.6	80.8±0.4/0.073±5e-3	88.6±0.4	74.8±2.9	88.2±0.7	88.6±0.3	88.6±0.4/0.086±4e-3
CSST	77.5±1.5	72.1±0.2	76.5±4.9	76.8±5.2	77.5±1.5/0.091±3e-3	87.6±0.7	78.1±0.3	86.1±0.7	87.1±0.2	88.7±0.1/0.093±5e-4
FixMatch (LA)	79.4±0.7	61.1±1.2	78.3±0.8	78.7±1.1	79.4±0.7/0.064±1e-3	89.8±0.3	75.9±1.6	89.3±0.2	89.6±0.1	89.8±0.3/0.086±1e-3
w/SelMix (Ours)	**81.3**±0.5	**74.3**±1.2	**81.0**±0.8	**81.1**±0.5	**81.7**±0.8/**0.091**±3e-3	**91.4**±1.2	**84.7**±0.7	**91.1**±1.1	**91.3**±1.2	**91.4**±1.2/**0.096**±9e-4

Table O.4. Comparison across methods when label distribution ρ_u is unknown. We use the STL-10 dataset for comparison.

	STL-10 ($\rho_l = 10, \rho_u = \text{NA}, N_1 = 450, \sum_i M_i = 100\text{k}$)				
	Mean Rec.	Min Rec.	HM	GM	Mean Rec./Min Cov.
DARP	76.5±0.3	54.7±1.9	74.0±0.5	75.3±0.4	76.5±0.3/0.058±2e-3
CReST	70.1±0.3	48.2±2.2	67.1±1.1	67.8±1.1	70.1±0.3/0.066±2e-3
DASO	78.1±0.5	55.8±3.7	76.6±1.1	77.2±0.2	78.1±0.5/0.083±3e-3
ABC	77.5±0.4	55.4±6.7	74.7±1.5	76.3±0.9	77.5±0.4/0.079±7e-3
CSST	79.2±1.5	50.8±2.9	78.3±2.6	78.9±2.1	79.2±1.5/0.081±6e-3
FixMatch(LA)	77.9±1.1	52.2±4.1	74.6±1.7	76.1±1.4	77.9±1.1/0.066±5e-3
w/SelMix (Ours)	**80.9**±0.5	**68.5**±1.8	**79.1**±1.2	**80.1**±0.4	**80.9**±0.5/**0.088**±1e-3

Training Details. Our classifier h is composed of a feature extractor $g : \mathcal{X} \to \mathbb{R}^d$ followed by a linear softmax classification layer $f : \mathbb{R}^d \to \Delta_{K-1}$, as mentioned in Sec. 2. As our classifier, we use the Wide ResNet-28-2 (Zagoruyko & Komodakis, 2016) pretrained using FixMatch (Sohn et al., 2020) with the cross-entropy loss replaced by the logit adjusted (LA) cross-entropy loss (Menon et al., 2020). This replacement helps generate pseudo-labels debiased from the long-tail imbalance of the training set (Appendix N). This results in better consistency regularization, hence a better feature extractor g, especially in cases where there is a mismatch of label distribution of labeled and unlabeled data. We perform fine-tuning of the model through SelMix (Alg. 1) using cosine learning rate and SGD optimizer, we freeze the batchnorm layers and fine-tune the feature extractor with low learning rate. This is done to ensure that the mean feature statistics z_k don't change much during the process, as per our theoretical results. The appendix mentions additional details and hyperparameter values in Table J.1.

Evaluation Setup. For evaluation, we compare the results of our work to the state-of-the-art empirical baselines of CReST, CreST+ (Wei et al., 2021), DASO (Oh et al., 2022), DARP (Kim et al., 2020a) and ABC (Lee et al., 2021) in semi-supervised long-tailed learning. We consider the public codebase of DASO (Oh et al., 2022) for all these baselines and report the results corresponding to the SotA base pre-training method of FixMatch + LA. We also compare with CSST (Rangwani et al., 2022), which is a theoretically principled method by obtaining the implementation from authors. We compare the methods on two broad sets of metric objectives: **a) Unconstrained objectives** which includes G-mean, H-mean, Mean (Arithmetic Mean), and worst-case (Min.) Recall **b) Constrained objectives** include maximizing the recalls under coverage constraints. The constraint for all classes is that coverage should be greater than $\frac{0.95}{K}$. As followed in the literature (Narasimhan & Menon, 2021) for CIFAR-100 due to its small size, instead of Min Recall/Coverage, we optimize the Min Head-Tail Recall/Min Head-Tail coverage, respectively. The tail corresponds to the least frequent ten classes, and the head corresponds to the other 90 classes. For a more detailed overview of the metric objectives and their definition, refer to Table J.3. We report mean and std. deviation of results across 3 seeds.

O.2. Results on Matched Label Distributions

In this section, we report results for the typical case of $\rho_l = \rho_u$, i.e., matched unlabeled and labeled distributions of class labels. We report the metrics for Mean, G-mean, H-mean, and worst-case (Min) recall for all baselines in Table O.5. For reporting results for CSST and SelMix, we report the results after training them for the specified metric. We observe significant improvement in the corresponding metric when FixMatch (LA) model is fine-tuned with the proposed SelMix method in all cases. For the metric of Min Recall, a 5% improvement is observed for CIFAR-10, and a corresponding 9.8 % improvement in Min HT Recall for CIFAR-100 over SotA methods which aim for optimizing accuracy. We also find

a significant improvement over SotA in the H-mean of Recall for CIFAR-100, showing the reduction in the performance disparity of the model between the head and tail classes. Further, even for the standard metric of mean recall ($\sim$ accuracy), SelMix achieves improved performance compared to baselines. We find that for optimizing the $\mathcal{P}_{\text{SelMix}}$, initially mixes up samples from tail classes to increase performance on them and then gradually moves towards uniform mixups in the later part (Appendix L).

We now move towards focusing on the optimization of metrics while satisfying the coverage constraints (i.e., $\text{cov}_\text{i}[h] \geq \frac{0.95}{K}$). We first focus on optimizing the model's mean recall with coverage constraint, as CSST supports it. However, as SelMix is generic, it also supports optimizing metrics like H-mean with coverage, which we show in the Appendix K. In Table O.5, we find that most heuristic SotA methods lead to sub-optimal min. coverage values, and only CSST and SelMix approximately satisfy the coverage constraints (underlined). Further, SelMix produces better mean recall performance than CSST, providing a better tradeoff in terms of performance and satisfiability of constraints. The models obtained after improving on such constrained objectives often perform well across all metrics and lead to more balanced and fair models . We find that this observation holds true even in the detailed analysis of SelMix models across metrics presented in Appendix M.

Optimising constrained NDO: In our setup, metrics such as Minimum Recall and fairness metrics such as Mean Recall s.t. per-class coverage $\geq \frac{0.95}{K}$ and H-mean s.t. per-class coverage $\geq \frac{0.95}{K}$ observe the most significant gains in performance, for CIFAR-10 SelMix and CSST are the only methods that satisfy the coverage constraint while optimising mean recall and H-mean. This is due to the inherent design of these methods capable of optimising such constrained Non-Decomposable objectives and SelMix improves over CSST w.r.t the underlying objective with a significant margin. This is true for CIFAR-100 too where we optimise for Mean Recall and H-mean subject to the constraint of the minimum of head class and tail class coverages being $\geq \frac{0.95}{K}$. Here too, other methods which are often designed w.r.t overall accuracy fail to satisy the coverage constraints and SelMix that not only satisfies them but also has either competitive or superior Mean Recall and H-mean. SelMix shows the biggest performance gain in optimising for Min. Recall for CIFAR-10 and Min. of Head class and Tail class recall for CIFAR-100. This is because when a certain class suffers in recall, the sampling function $\mathcal{P}^t$ dynamically adjusts to oversample from those classes from the labeled set, which in-turn increses its recall.

Table O.5. We compare the results for CIFAR-10 and CIFAR-100 when the labeled and unlabeled data distribution matches, this is the classical imbalanced semi-supervised learning case. CIFAR-10 is kept at an imbalance ratio $\rho = 100$ while CIFAR-100 is kept at $\rho = 10$

	CIFAR-10 ($\rho_l = \rho_u = 100, N_1 = 1500, M_1 = 3000$)						**CIFAR-100** ($\rho_l = \rho_u = 10, N_1 = 150, M_1 = 300$)					
	Mean Rec.	Min Rec.	HM	GM	HM/Min Cov.	Mean Rec./Min Cov.	Mean Rec.	Min H-T Rec.	HM	GM	HM/Min H-T Cov.	Mean Rec./Min H-T Cov.
Fixmatch + LA	82.6$_{\pm 1.1}$	63.6$_{\pm 6.3}$	81.1$_{\pm 1.5}$	81.8$_{\pm 1.3}$	82.6$_{\pm 1.1}$/0.065$_{\pm 8e\text{-}3}$	82.6$_{\pm 1.1}$/0.065$_{\pm 8e\text{-}3}$	58.8$_{\pm 0.1}$	36.2$_{\pm 1.1}$	45.0$_{\pm 0.5}$	53.2$_{\pm 0.1}$	45.0$_{\pm 0.5}$/0.0055$_{\pm 2e\text{-}4}$	58.8$_{\pm 0.1}$/0.0055$_{\pm 2e\text{-}4}$
w/ DARP	83.3$_{\pm 0.4}$	66.4$_{\pm 3.1}$	81.9$_{\pm 0.5}$	82.6$_{\pm 0.4}$	83.3$_{\pm 0.4}$/0.070$_{\pm 3e\text{-}3}$	83.3$_{\pm 0.4}$/0.070$_{\pm 3e\text{-}3}$	60.1$_{\pm 0.2}$	39.6$_{\pm 1.1}$	48.7$_{\pm 1.3}$	55.4$_{\pm 0.5}$	48.7$_{\pm 1.3}$/0.0040$_{\pm 2e\text{-}3}$	60.1$_{\pm 0.3}$/0.0040$_{\pm 2e\text{-}3}$
w/ CReST	82.1$_{\pm 0.6}$	68.2$_{\pm 3.2}$	81.0$_{\pm 0.7}$	81.6$_{\pm 0.7}$	82.1$_{\pm 0.6}$/0.073$_{\pm 5e\text{-}3}$	82.1$_{\pm 0.6}$/0.073$_{\pm 5e\text{-}3}$	58.2$_{\pm 0.2}$	40.7$_{\pm 0.7}$	48.3$_{\pm 0.2}$	54.1$_{\pm 0.1}$	48.3$_{\pm 0.2}$/0.0083$_{\pm 2e\text{-}4}$	58.2$_{\pm 0.3}$/0.0083$_{\pm 2e\text{-}4}$
w/CReST+	83.1$_{\pm 0.3}$	71.3$_{\pm 1.5}$	82.2$_{\pm 0.2}$	82.6$_{\pm 0.3}$	83.1$_{\pm 0.3}$/0.076$_{\pm 2e\text{-}3}$	83.1$_{\pm 0.3}$/0.076$_{\pm 2e\text{-}3}$	57.8$_{\pm 0.8}$	42.1$_{\pm 0.7}$	48.2$_{\pm 0.6}$	53.8$_{\pm 0.9}$	48.2$_{\pm 0.6}$/0.0088$_{\pm 1e\text{-}4}$	57.8$_{\pm 0.8}$/0.0088$_{\pm 1e\text{-}4}$
w/DASO	84.1$_{\pm 0.3}$	72.6$_{\pm 2.1}$	83.5$_{\pm 0.3}$	83.8$_{\pm 0.3}$	84.1$_{\pm 0.3}$/0.083$_{\pm 1e\text{-}3}$	84.1$_{\pm 0.3}$/0.083$_{\pm 1e\text{-}3}$	60.6$_{\pm 0.2}$	40.9$_{\pm 0.4}$	49.1$_{\pm 0.7}$	55.9$_{\pm 0.1}$	49.1$_{\pm 0.7}$/0.0063$_{\pm 3e\text{-}4}$	60.6$_{\pm 0.3}$/0.0063$_{\pm 3e\text{-}4}$
w/ABC	85.1$_{\pm 0.5}$	74.1$_{\pm 0.6}$	84.6$_{\pm 0.5}$	84.9$_{\pm 0.6}$	85.1$_{\pm 0.5}$/0.086$_{\pm 3e\text{-}3}$	85.1$_{\pm 0.5}$/0.086$_{\pm 3e\text{-}3}$	59.7$_{\pm 0.2}$	46.4$_{\pm 0.6}$	50.1$_{\pm 1.2}$	55.6$_{\pm 0.4}$	50.1$_{\pm 1.2}$/0.0089$_{\pm 1e\text{-}3}$	59.7$_{\pm 0.3}$/0.0089$_{\pm 1e\text{-}3}$
FixMatch + w/CSST	81.1$_{\pm 0.2}$	71.7$_{\pm 0.2}$	76.9$_{\pm 0.2}$	77.7$_{\pm 0.7}$	81.1$_{\pm 0.2}$/0.090$_{\pm 2e\text{-}4}$	81.1$_{\pm 0.2}$/0.090$_{\pm 2e\text{-}4}$	57.2$_{\pm 0.2}$	48.4$_{\pm 0.3}$	47.7$_{\pm 0.8}$	53.5$_{\pm 0.4}$	47.7$_{\pm 0.8}$/0.0096$_{\pm 4e\text{-}3}$	57.2$_{\pm 0.3}$/0.0096$_{\pm 4e\text{-}3}$
w/SelMix (Ours)	85.4$_{\pm 0.1}$	79.1$_{\pm 0.1}$	85.1$_{\pm 0.1}$	85.3$_{\pm 0.1}$	84.9$_{\pm 0.2}$/0.094$_{\pm 3e\text{-}4}$	85.7$_{\pm 0.2}$/0.095$_{\pm 1e\text{-}3}$	59.8$_{\pm 0.2}$	57.8$_{\pm 0.5}$	53.8$_{\pm 0.5}$	56.7$_{\pm 0.4}$	53.8$_{\pm 0.5}$/0.0098$_{\pm 5e\text{-}5}$	59.6$_{\pm 0.2}$ / 0.0097$_{\pm 1e\text{-}4}$