Last-Layer Fairness Fine-tuning is Simple and Effective for Neural Networks

Yuzhen Mao [1] **Zhun Deng** [2] **Huaxiu Yao** [3] **Ting Ye** [4] **Kenji Kawaguchi** [5] **James Zou** [3]

Abstract

As machine learning has been deployed ubiquitously across applications in modern data science, algorithmic fairness has become a great concern. Among them, imposing fairness constraints during learning, i.e. in-processing fair training, has been a popular type of training method because they don't require accessing sensitive attributes during test time in contrast to post-processing methods. While this has been extensively studied in classical machine learning models, their impact on deep neural networks remains unclear. Recent research has shown that adding fairness constraints to the objective function leads to severe over-fitting to fairness criteria in large models, and how to solve this challenge is an important open question. To tackle this, we leverage the wisdom and power of pre-training and fine-tuning and develop a simple but novel framework to train fair neural networks in an efficient and inexpensive way — last-layer fine-tuning alone can effectively promote fairness in deep neural networks. This framework offers valuable insights into representation learning for training fair neural networks. The code is published at https://github.com/yuzhenmao/Fairness-Finetuning

1. Introduction

The social impacts of machine learning systems deployed in our daily lives are getting increasing attention in modern data science. Great efforts have been put into understanding and correcting biases in algorithms (Hardt et al., 2016; Dwork et al., 2012). However, most of the research has been conducted on understanding and correcting biases in classical machine learning models and simple datasets such as regression models and the adult income dataset (Ding et al., 2021). In contrast, in modern data science, tasks are more complex (e.g. classification on high dimensional datasets such as images), and over-parameterized models such as neural networks are deployed, which have been proven to reach the state-of-art in prediction performance. Thus, it is critical to study and understand how fairness techniques work on modern architectures.

Among all fairness techniques, in-processing fair training, which imposes fairness constraints during learning, have been a popular type of fair training method given the advantage of not requiring to access sensitive attributes during test time as post-processing methods (Kim et al., 2019; Hardt et al., 2016) and can more efficiently use the information of labels compared with pre-processing methods (Madras et al., 2018). However, as pointed out by Cherepanova et al. (2021) in-processing techniques are less effective for over-parameterized large neural networks because the model can easily overfit the fairness objectives during training, especially when the training data is imbalanced. Cherepanova et al. (2021) raised this fairness over-fitting issue as an open challenge. Although there has been work (Deng et al., 2022b) trying to address the challenge, the computational cost is much more expensive compared to standard training.

In this paper, we aim to tackle the challenge mentioned above and avoid the issue of overfitting the fairness criteria in an *efficient and inexpensive* way when training neural networks. We focus on pre-training and fine-tuning, which have been proven to be useful techniques in obtaining powerful neural networks and are widely applied in state-of-the-art object detection. According to Kirichenko et al. (2022), re-training the last layer of suitably pre-trained representations can reduce vulnerability to spurious correlation, and thus significantly improve prediction accuracy on imbalanced dataset and model robustness to covariate shift. A natural question induced by that is:

"*Will fine-tune the last layer or a small fraction of a standard-trained neural network with fair training methods be enough to obtain a fair neural network?*"

We provide a positive answer to the above question. As our *main contribution*, we leverage the wisdom and power of pre-training and fine-tuning so as to develop a simple but effective framework to train fair neural networks in an

[1]Simon Fraser University [2]Harvard University [3]Stanford University [4]University of Washington [5]National University of Singapore. Correspondence to: Zhun Deng <zhundeng@g.harvard.edu>, James Zou <jamesz@stanford.edu>.

Published at the ICML 2023 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA.

efficient and inexpensive way (as illustrated in Figure 1 in the appendix): (1) pre-training to obtain a representation by standard empirical risk minimization; (2) fine-tuning a few extra layers of neural networks by imposing fairness constraints while fixing the obtained representation (more details in Section 4). In addition, we further show that our method can even work for out-of-domain data by fine-tuning while we only train the representation on a source dataset. Finally, we also explore whether fine-tuning other structures beyond the last layer of neural networks can perform well (see details in Section C.3).

2. Preliminaries

2.1. Notation

We consider a dataset consists of triplets, i.e. $\mathcal{D} = \{(x_i, a_i, y_i)\}_{i=1}^N$, where for each triplet, x_i is the feature vector drawn from an input distribution over $\mathcal{X}$, $a_i \in \mathcal{A}$ denotes the corresponding sensitive attribute such as race or gender, and $y_i \in \mathcal{Y}$ is the corresponding label. Throughout the paper, for simplicity, we only consider a and y as binary variables, where $a \in \{0, 1\}$ and $y \in \{0, 1\}$. However, our method can easily be generalized to multiple sensitive attributes and multi-class scenarios. We further denote the cross entropy loss as $\hat{L}(h) = -\sum_{i=1}^m y_i \cdot \log\left(p(h(x_i))\right)/m$, where $p \circ h$ is the estimation of the prediction probability for the correct class for a sample x_i, and p is a soft-max function.

2.2. Fairness notions

In standard supervised learning tasks, people aim to train a model $h \in \mathcal{H} : \mathcal{X} \mapsto \mathcal{Y}$, where $\hat{y}_i = h(x_i)$ is the prediction of y_i for a given feature vector x_i. For example, in the CelebA dataset (Liu et al., 2015), the task is to classify the hair color of the celebrity in the image and use gender as the sensitive attribute. In CelebA dataset, there are four different groups corresponding to four different combinations of (y_i, a_i): Blonde woman ($\mathcal{G}_1$), blonde man ($\mathcal{G}_2$), Non-blonde woman ($\mathcal{G}_3$) and Non-blonde man ($\mathcal{G}_4$). Since $\mathcal{G}_2$ contains only 1% images of the whole dataset, it is referred as the *minority group*. For larger groups such as $\mathcal{G}_1$ and $\mathcal{G}_4$, we refer them as *majority groups*. This imbalance existing in the dataset usually results in an unfairly biased model with standard training. To de-bias the model and make it "fairer", people propose adding varieties of additional fairness constraints (regularization terms) to objective functions to achieve model fairness. Specifically, in this paper, we mainly focus on the following three popular fairness notions which have been widely used in the previous literature: Equalized Odds (EO), Accuracy Equality (AE) and Max-Min Fairness (MMF). In this paper, we don't discuss notions such as Disparate Impact or Demographic Parity because their definitions are problematic in the way that they cannot distinguish qualified individuals from others in each group (Hardt et al., 2016) and in general cannot align with the model accuracy well.

Equalized odds (EO). Equalized odds requires, given the true label y, an algorithm's decisions/outcomes do not depend on the sensitive attributes of individuals, such as race, gender, or age, which indicates that $\hat{y}$ is conditionally independent of the sensitive attribute a given y. In other words, it means that the false positive rate and false negative rate should be the same for all groups, so that no group is unfairly disadvantaged. Equalized odds (Hardt et al., 2016) is defined as:

$$\mathbb{P}(\hat{y} = 1 \mid a = 0, y = y) = \mathbb{P}(\hat{y} = 1 \mid a = 1, y = y) \quad (1)$$

To enforce equalized odds in practice, one way people implement in training is to minimize the following objectives (Manisha & Gujar, 2018; Cherepanova et al., 2021):

$$\min_h \left[\hat{L}_w(h) + \alpha(fpr + fnr)\right] \quad (2)$$

for a given predefined weight α, where

$$fpr = \left|\frac{\sum_i p_i(1-y_i)a_i}{\sum_i a_i} - \frac{\sum_i p_i(1-y_i)(1-a_i)}{\sum_i (1-a_i)}\right|$$
$$fnr = \left|\frac{\sum_i (1-p_i)y_i a_i}{\sum_i a_i} - \frac{\sum_i (1-p_i)y_i(1-a_i)}{\sum_i (1-a_i)}\right|.$$

Here, p_i denotes a softmax output (binary prediction task) of the model h and $\hat{L}_w$ is the weighted cross-entropy loss.

Accuracy equality (AE). Accuracy equality requires an algorithm produces outcomes that are (approximately) equally accurate for individuals belonging to different protected groups. Its goal is to ensure that an algorithm does not unfairly advantage or disadvantage certain groups, and instead provides equally accurate predictions for all individuals. In other words, a model satisfies accuracy equality if its misclassification rates are equal across different sensitive groups (Zafar et al., 2017). Accuracy equality is defined as:

$$\mathbb{P}(\hat{y} \neq y \mid a = 0) = \mathbb{P}(\hat{y} \neq y \mid a = 1) \quad (3)$$

To enforce accuracy equality in practice, one way people implement in training is to minimize the following objective:

$$\min_h \left[\hat{L}_w(h) + \alpha\left|\hat{L}^{a+}(h) - \hat{L}^{a-}(h)\right|\right] \quad (4)$$

for a given predefined weight α, where $\hat{L}^{a+}(h)$ is the cross entropy loss of samples with $a = 1$, and $\hat{L}^{a-}(h)$ is the cross entropy loss of samples with $a = 0$. $\hat{L}_w$ is the weighted cross-entropy loss.

Max-Min fairness (MMF). Max-min fairness focuses on maximizing the performance of the worse-off group,

i.e., the group with the lowest utility (Lahoti et al., 2020; Cherepanova et al., 2021). It is defined as:

$$\max \min_{y\in\mathcal{Y},a\in\mathcal{A}} \mathbb{P}(\hat{y}=y \mid y,a) \tag{5}$$

To satisfy max-min fairness, people aim to minimize the following objective (Cherepanova et al., 2021):

$$\min_h \max\{\hat{L}^{(y+,a+)}(h), \hat{L}^{(y+,a-)}(h), \hat{L}^{(y-,a+)}(h), \hat{L}^{(y-,a-)}(h)\}, \tag{6}$$

where $\hat{L}^{(y',a')}(h)$ denotes the cross-entropy loss on the training samples where $y=y'$ and $a=a'$.

3. Problem Background

3.1. Challenges in training fair neural networks with in-processing techniques

To tackle fairness concerns in modern data science, researchers have proposed and formalized various notions of fairness as well as methods for mitigating unfair behavior. However, as pointed out by Cherepanova et al. (2021), Deng et al. (2022b) and our introduction section, the effectiveness of in-processing techniques in fairness that impose fairness constraints on modern structures such as deep neural networks is unclear and still under exploration. Specifically, Cherepanova et al. (2021) observe that large models overfit to fairness objectives and produce a range of unintended and undesirable consequences by conducting studies on both facial recognition and automated medical diagnosis datasets using state-of-the-art architectures. They empirically emphasize the over-fitting issue in training fair neural networks, where models trained with fairness constraints become too closely aligned with the training data, leading to poor performance on unseen data in terms of fairness.

Their studies are mainly based on two main approaches for rectifying unfair behavior. (1) The first one is to impose the fairness constraints or regularizers on the training objective and train the *full* neural network. Based on the experiments, they find the model shows excellent performance on the training set and appears to be fair, in terms of the difference in AUC (see details in Section 5) values for different sensitive attributes. However, upon evaluation on the test set, models trained with fairness constraints can be proven to be even less fair compared to a baseline model, which indicates a very serious over-fitting issue. Increasing the strength of the constraints results in a higher accuracy trade-off, but it still fails to significantly improve fairness on the validation and test sets. Cherepanova et al. attribute this to the over-parameterized nature of deep neural networks and the fluid decision boundary it creates. (2) They also try to only apply fairness penalties on a holdout set after training a model without fairness constraints and fine-tune the *full* neural network on the hold-out set. They assume that the ineffectiveness of fairness constraints on the training set is due to the high training accuracy, which makes the models appear fair regardless of their performance during testing. However, the issue of over-fitting remains prevalent, that is, the fairness is achieved on the train dataset but cannot be generalized to unseen data.

3.2. Our Inspiration: standard training can still learn core features on imbalanced datasets

In addition to the previous success of pre-training and fine-tuning in mitigating the issue of spurious correlation (Kirichenko et al., 2022), our inspiration is also drawn from the observation that core features can still be learned by standard training on imbalanced datasets, which can be used to enable accurate predictions for minority groups in the later fine-tuning phase.

To access this property, we evaluate the ResNet-18 model that has been trained on the original CelebA dataset using empirical risk minimization (ERM), on a customized hair-only dataset D_H, which contains 29,300 sampled images of the hair segmented from the original training images using the mask from Lee et al. (2020) on uniform grey background. In order to effectively examine whether the model has learned the core features that are relevant to the labels and whether these features have been properly encoded in the preceding layers before the final layer, we divide the D_H into two sets: D_H^{Tr} and D_H^{Te}. D_H^{Tr} is evenly balanced and comprises 107 images from each (a, y) group, totaling 428 images. On the other hand, D_H^{Te} includes the remaining 28,872 images, with at least 107 images per group. Then we fine-tune the last layer of the model on D_H^{Tr} and evaluate it on D_H^{Te}. We repeat this process with the model pre-trained on a balanced dataset sampled from the original training dataset, as a comparison to the model pre-trained on the original imbalanced dataset. We present the mean and the worst group accuracy of all the experiments mentioned above in Table 1. Additionally, we also include the results on the original test dataset without fine-tuning for comparison purposes. Based on the results, we find that there is a significant discrepancy between the mean and worst group accuracy on the original test data, which means that the standard-trained full model is heavily influenced by the imbalanced training dataset, leading to poor performance on minority groups. On the other hand, we find the last-layer-fine-tuned model achieves very good performance on the hair-only dataset with 86% and 82% worst group accuracy. This results indicate that although the model trained on the imbalanced dataset under-performs on the minority groups, it can still learn the core features which are relevant to the classification task, and only need simple fine-tuning to perform accurate predictions on the core features. This conclusion also well supports the method we analyze in the following

sections.

Train	Test (Worst/Mean)	
	Original	Hair-only
Original	0.268/0.946	0.863/0.878
Balanced	0.789/0.835	0.827/0.843

Table 1. Representation learning on CelebA. The column Original corresponds to directly evaluate on the original test dataset. The column Hair-only corresponds to last-layer fine-tuned results on the sampled hair-only dataset D_H^{Te}.

4. Our Main Approach: Fair Deep Feature Reweighting

In Cherepanova et al. (2021), the authors point out over-parameterization of a neural network lead to the over-fitting of fairness criteria since over-parameterization makes the learned neural network's decision boundary highly flexible, and trying to fit the boundaries to meet fairness criteria for one attribute can negatively impact fairness with regards to another sensitive attribute. However,over-parameterization has been a key for neural networks to achieve high accuracy in prediction, especially for those neural networks designed to tackle challenging tasks. Inspired by previous work (Kirichenko et al., 2022) on spurious correlation, in this section, we show how to solve this dilemma in a simple way based on pre-training and fine-tuning, and we call our method *fair deep feature reweighting*. Our method not only provides answers to the challenges brought up by Cherepanova et al. (2021), but it is also surprisingly simple and computationally cheap. Our approach also reveals an interesting future direction for research on fairness.

Step 1: pre-train a representation. As discussed in Section 3.2 and previous works (Lee et al., 2022; Kirichenko et al., 2022), standard training by doing empirical risk minimization (ERM) is enough to obtain representations that capture core features of input in many cases. With this spirit, we first train a neural network $\mathcal{N}$ with ERM and obtain a representation Φ, which is the sub-neural-network from the first layer to the next to the last layer of $\mathcal{N}$, i.e. $\mathcal{N} = w \circ \Phi$, where w is the last layer.

Step 2: fine-tune the last layer with reweighting and fairness constraints. We then fix Φ and improve model fairness through last-layer fine-tuning by incorporating fairness constraints (Eq. 2 & 4 & 6) and data reweighting obtain a new last layer w^{new} and another neural network $\mathcal{N}^{new} = w^{new} \circ \Phi$. Specifically, for data reweighting, we first sample a small dataset D_r from the training dataset D and the validation dataset $\hat{D}$. Each (a, y) group in D_r has the same number of samples, where a and y represent the sensitive attribute and the label respectively. We then train w^{new} from scratch on the balanced dataset D_r with standard ERM and fairness constraints.

In summary, fair deep feature reweighting, which only requires updating the parameters of the last layer when training with fairness constraints, avoids the over-parameterization issue described in Cherepanova et al. (2021) and reduces the risk of over-fitting. In addition, with respect to fairness, data reweighting can also be particularly beneficial in correcting models that have been negatively impacted by imbalanced training datasets, where one class is heavily overrepresented compared to others. Our approach allows the model to better capture patterns in the data, leading to improved fairness in its performance.

Intuition behind our approach. As implied by our observation in Section 3.2, ERM can already encode the information of core features well in a representation and we only need to further compose the representation with a linear structure to recover those core features and used them to predict. Since the fairness criteria are imposed only at the fine-tuning phase, we will not suffer from over-fitting issue.

5. Experiments

5.1. Experimental Setup

In this section, we briefly discuss our experimental setup and put detailed descriptions in Appendix B.

Datasets and Hyperparameters. To evaluate the performance of our methods, we conduct a comparative analysis using various baselines on the CelebA and UTKFace datasets, which focus on facial recognition. Our model architecture is based on ResNet-18, as employed in previous studies (Cherepanova et al., 2021). During the fine-tuning process, we replace the last layer with a newly initialized layer and subsequently update only this layer while keeping the remaining model parameters fixed. To train our model, we utilize SGD with a momentum of 0.9 and weight decay of 5e-4. All hyperparameters are carefully tuned using a separate validation dataset, and we provide a summary of these hyperparameters in Table 5. We evaluate the model's fairness and prediction.

Compared methods. We define baseline methods and our proposed method FDR as follows:

FullFT-Reg: Impose the fairness constraints on the training objective and train the full neural network.

LastFT: Fine-tune the last layer of a pre-trained model on the imbalanced validation dataset (Kirichenko et al., 2022).

LastFT-RW: Fine-tune the last layer of a pre-trained model on the balanced dataset.

LastFT-Reg: Fine-tune the last layer of a pre-trained model on the validation dataset with fairness constraints.

FDR (**ours**): Fine-tune the last layer of a pre-trained model

on the balanced dataset with fairness constraint.

Metrics. To assess the accuracy of our model, we employ weighted accuracy (WACC) and Area under the ROC Curve (AUC) as evaluation metrics, as suggested by Fawcett (2004). For evaluating fairness, we employ the metrics Equalized Odds Difference (EO_Diff), Accuracy Equality Difference (AE_Diff), and Worst Accuracy (WA). Additionally, we introduce a novel metric, denoted as AF, that takes into account both accuracy and fairness. Please refer to Appendix B.2 for detailed descriptions of metrics.

5.2. Assessing Fairness on CelebA Dataset

We conducted an evaluation of various methods on the CelebA dataset (see the dataset details in Appendix B.1) with the goal of accurately predicting the hair color in each image. The results of three different fairness notions are presented in Table 2. Our experiments revealed the following findings:

- Fine-tuning only the last layer of a pre-trained model, as opposed to fine-tuning all layers (FullFT-Reg), led to significant improvements in all fairness metrics and reduced their generalization gap between the training and test datasets. This indicates that last layer fine-tuning effectively addresses the issue of overfitting.
- Last-layer methods that incorporated fairness constraints, i.e., LastFT-Reg and FDR, exhibited relatively lower test WACC compared to methods without such constraints. This suggests that adding fairness constraints may negatively impact the model's prediction accuracy.
- Among the evaluated methods, FDR demonstrated the best performance in both fairness metrics and the accuracy-fairness (AF) evaluation. This implies that last layer fine-tuning, combined with data reweighting and fairness constraints, can efficiently and effectively mitigate the overfitting issue.

Apart from the above results, we conduct experiments under transfer learning setting and using surgical fine-tuning (Lee et al., 2022) in Appendix C.2 and C.3, respectively. All results support our claims.

Table 2. Overall performance on CelebA dataset with different fairness constraints, where averaged values are reported over twenty random trials.

Fairness Notion 1: EO	WACC		AUC		EO_Diff		AF
	Train	Test	Train	Test	Train	Test	Test
FullFT-Reg	1.000	**0.914**	1.000	0.969	0.000	0.499	0.415
LastFT	0.918	0.913	0.974	**0.971**	0.308	0.327	0.586
LastFT-RW	0.913	0.908	0.970	0.968	0.100	0.207	0.701
LastFT-Reg	0.898	0.901	0.971	0.969	0.177	0.153	0.748
FDR	0.898	0.892	0.962	0.958	0.031	**0.107**	**0.785**
Fairness Notion 2: AE	WACC		AUC		AE_Diff		AF
	Train	Test	Train	Test	Train	Test	Test
FullFT-Reg	1.000	**0.914**	1.000	0.968	0.000	0.049	0.865
LastFT	0.918	0.913	0.974	**0.971**	0.066	0.043	0.869
LastFT-RW	0.913	0.908	0.970	0.968	0.026	0.020	0.888
LastFT-Reg	0.907	0.904	0.969	0.964	0.016	0.009	0.895
FDR	0.898	0.900	0.963	0.967	0.009	**0.003**	**0.897**
Fairness Notion 3: MMF	WACC		AUC		WA		AF
	Train	Test	Train	Test	Train	Test	Test
FullFT-Reg	1.000	0.910	1.000	0.969	1.000	0.393	1.303
LastFT	0.918	**0.913**	0.974	**0.971**	0.633	0.598	1.511
LastFT-RW	0.913	0.908	0.970	0.968	0.872	0.732	1.640
LastFT-Reg	0.896	0.888	0.960	0.955	0.879	0.717	1.605
FDR	0.902	0.898	0.964	0.962	0.868	**0.803**	**1.701**

6. Conclusion

In this paper, we present a simple yet innovative framework for training fair neural networks through the use of pre-training and fine-tuning. The experimental findings compellingly illustrate that fine-tuning the last layer alone with data reweighting is sufficient for promoting fairness in deep neural networks.

References

Binns, R. On the apparent conflict between individual and group fairness. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pp. 514–524, 2020.

Burhanpurkar, M., Deng, Z., Dwork, C., and Zhang, L. Scaffolding sets. *arXiv preprint arXiv:2111.03135*, 2021.

Cherepanova, V., Nanda, V., Goldblum, M., Dickerson, J. P., and Goldstein, T. Technical challenges for training fair neural networks. *arXiv preprint arXiv:2102.06764*, 2021.

Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

Deng, Z., Ding, F., Dwork, C., Hong, R., Parmigiani, G., Patil, P., and Sur, P. Representation via representations: Domain generalization via adversarially learned invariant representations. *arXiv*, 2006.11478, 2020.

Deng, Z., He, H., and Su, W. Toward better generalization bounds with locally elastic stability. In *International Conference on Machine Learning*, pp. 2590–2600. PMLR, 2021a.

Deng, Z., Zhang, L., Ghorbani, A., and Zou, J. Improving adversarial robustness via unlabeled out-of-domain data. In *International Conference on Artificial Intelligence and Statistics*, pp. 2845–2853. PMLR, 2021b.

Deng, Z., Zhang, L., Vodrahalli, K., Kawaguchi, K., and Zou, J. Y. Adversarial training helps transfer learning via better representations. *Advances in Neural Information Processing Systems*, 34, 2021c.

Deng, Z., Sun, H., Wu, Z. S., Zhang, L., and Parkes, D. C. Reinforcement learning with stepwise fairness constraints. *arXiv preprint arXiv:2211.03994*, 2022a.

Deng, Z., Zhang, J., Zhang, L., Ye, T., Coley, Y., Su, W. J., and Zou, J. Fifa: Making fairness more generalizable in classifiers trained on imbalanced data. *arXiv preprint arXiv:2206.02792*, 2022b.

Ding, F., Hardt, M., Miller, J., and Schmidt, L. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems*, 34:6478–6490, 2021.

Dwork, C. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.

Fawcett, T. Roc graphs: Notes and practical considerations for researchers. *Machine learning*, 31(1):1–38, 2004.

Gerstenmayer, A. and Jüngel, A. Analysis of a degenerate parabolic cross-diffusion system for ion transport. *Journal of Mathematical Analysis and Applications*, 461(1): 523–543, 2018.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Ji, W., Deng, Z., Nakada, R., Zou, J., and Zhang, L. The power of contrast for feature learning: A theoretical analysis. *arXiv*, 2110.02473, 2021a.

Ji, W., Lu, Y., Zhang, Y., Deng, Z., and Su, W. J. An unconstrained layer-peeled perspective on neural collapse. *arXiv*, 2110.02796, 2021b.

Kamiran, F. and Calders, T. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33, 2012.

Kawaguchi, K., Deng, Z., Luh, K., and Huang, J. Robustness implies generalization via data-dependent generalization bounds. In *International Conference on Machine Learning*, pp. 10866–10894. PMLR, 2022a.

Kawaguchi, K., Zhang, L., and Deng, Z. Understanding dynamics of nonlinear representation learning and its application. *Neural Computation*, 34(4):991–1018, 2022b.

Kawaguchi, K., Deng, Z., Ji, X., and Huang, J. How does information bottleneck help deep learning? *arXiv preprint arXiv:2305.18887*, 2023.

Kim, M. P., Ghorbani, A., and Zou, J. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 247–254, 2019.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

Lahoti, P., Beutel, A., Chen, J., Lee, K., Prost, F., Thain, N., Wang, X., and Chi, E. Fairness without demographics through adversarially reweighted learning. *Advances in neural information processing systems*, 33:728–740, 2020.

Lee, C.-H., Liu, Z., Wu, L., and Luo, P. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P., and Finn, C. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pp. 3384–3393. PMLR, 2018.

Manisha, P. and Gujar, S. Fnnc: achieving fairness through neural networks. *arXiv preprint arXiv:1811.00247*, 2018.

Park, S., Kim, D., Hwang, S., and Byun, H. Readme: Representation learning by fairness-aware disentangling method, 2020.

Petersen, F., Mukherjee, D., Sun, Y., and Yurochkin, M. Post-processing for individual fairness. *Advances in Neural Information Processing Systems*, 34:25944–25955, 2021.

Zafar, M. B., Valera, I., Rogriguez, M. G., and Gummadi, K. P. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pp. 962–970. PMLR, 2017.

Zhang, L., Deng, Z., Kawaguchi, K., Ghorbani, A., and Zou, J. How does mixup help with robustness and generalization? *arXiv*, 2010.04819, 2020.

Zhang, Z., Song, Y., and Qi, H. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5810–5818, 2017.

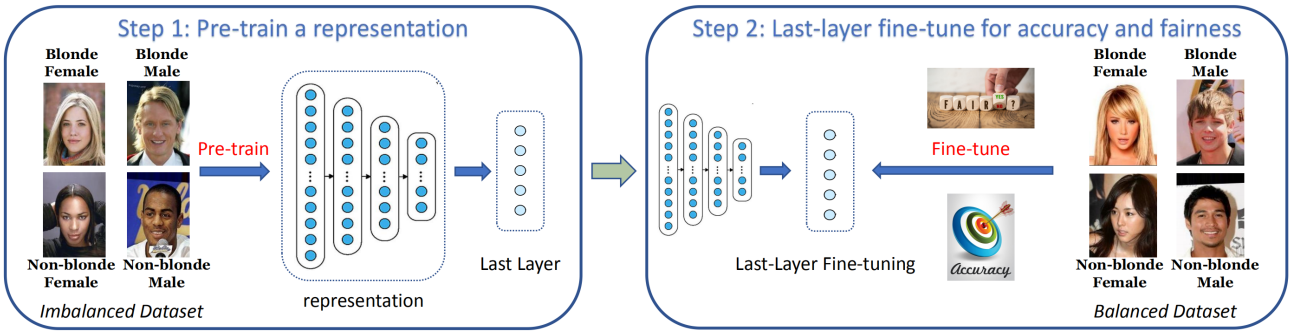

Figure 1. **Scheme of our approach.** We propose a simple framework to obtain fair neural networks by in-processing techniques. (1) We first obtain a representation by training a model via standard empirical risk minimization on the dataset (possibly imbalanced). (2) Next, we only fine-tune the last layer of the model on a balanced version of the original dataset via varities of in-processing techniques in fairness.

A. Additional Related work

Fairness in machine learning. Fairness in machine learning has been the focus of much research in recent years. A growing body of literature has aimed at addressing the potential biases in machine learning models with respect to sensitive attributes such as race, gender, and age (Zafar et al., 2017; Kamiran & Calders, 2012; Hardt et al., 2016). Researchers have proposed various fairness metrics and algorithms to mitigate such biases, including group fairness (Dwork et al., 2012; Binns, 2020; Deng et al., 2022a), individual fairness (Dwork, 2011; Petersen et al., 2021), and causal fairness (Gerstenmayer & Jüngel, 2018). Another existing approach involves modifying the learning algorithm to ensure that the machine learning model is fair with respect to certain protected groups (Zafar et al., 2017). This has led to the development of various algorithmic fairness techniques, such as data reweighting, adversarial debiasing, and regularization-based methods. The challenge lies in balancing fairness and accuracy, as the former can sometimes come at the cost of the latter (Chouldechova, 2017).

Last layer fine-tuning. Fixed-feature learning (Deng et al., 2020; 2021c; Ji et al., 2021b; Deng et al., 2021b; Burhanpurkar et al., 2021; Ji et al., 2021a; Kawaguchi et al., 2022b) has been popular for its efficiency to adopt to out-of-domain data. Fine-tuning only a small fraction of the parameters can also effectively avoid over-fitting and bad generalization (Deng et al., 2021a; Kawaguchi et al., 2022a; Zhang et al., 2020; Kawaguchi et al., 2023). Previous works (Lee et al., 2022; Kirichenko et al., 2022) have shown that instead of updating all the parameters of the model, only fine-tuning the last layer(s) can still match or even achieve better performance for spurious correlation and distribution shifts. Specifically, this approach involves training the last layer of a pre-trained neural network on a smaller, more specific dataset, with the aim of fine-tuning the model to perform well on this specific task. By using a pre-trained network as a starting point, last layer fine-tuning reduces the risk of over-fitting, as the initial layers have already learned high-level features from a large dataset. Another advantage of last-layer fine-tuning is that it can be time and computational resources efficient compared with training a network from scratch, since the pre-trained weights provide a good initialization point, allowing the network to converge faster.

B. Experiment Setup

B.1. Dataset Description

We mainly conduct our experiments on two popular datasets in facial recognition: CelebA (Liu et al., 2015) and UTK-Face (Zhang et al., 2017). Due to the page limit, we only show the results of CelebA in the main text, and present the results of UTKFace in Section C. Both datasets are of high dimension and have been widely used in prior deep learning and fairness studies. We provide a comprehensive description of the datasets and the corresponding tasks as follows. We create the following four subsets for each dataset – training set, validation set, test set, and an additional balanced dataset.

- **CelebA**: a large-scale dataset of celebrity faces, including over 200k images with annotations of 40 different attributes such as facial landmarks, gender, age, hair color, glasses, etc (Liu et al., 2015). In the experiment, we select hair color (blonde or not) as the label y to predict, and use gender (male or not) as the sensitive attribute a. Specifically, (male,

blonde hair) which only contains 1% of the total images, is the minority group of this dataset. We follow the dataset splitting format in the orignal paper and list the details in Tabel 3. We construct a balanced sub-dataset by sampling from the original training and validation datasets based on the number of images in the minority group. Specifically, we select 1,569 images per (y, a) group, yielding a total of 6,276 images in the balanced dataset.

- **UTKFace**: a large, publicly available face dataset with long age span (range from 0 to 116 years old) (Zhang et al., 2017). In our experiment, we randomly select 20% data (maintain the same proportion) from the training dataset to serve as the validation dataset and randomly select another 20% data (maintain the same proportion) from the training dataset to serve as the test dataset. The details of UTKFace dataset is presented in Tabel 4. We select gender as the sensitive attribute and age as the label to predict. Following the setting of prior work (Park et al., 2020), the age feature is annotated into young (≤ 35) and the others (> 35). In UTKFace, we also create a balanced sub-dataset by sampling an equal number of images from the original training and validation datasets for each (y, a) group. Specifically, we select 2,477 images per group, resulting in a total of 9,908 images in the balanced dataset.

Table 3. CelebA dataset statistics.

(train/val/test)	Blonde Hair	Non-blonde Hair	Total
Male	1,387/182/180	66,874/8,276/7,535	68,261/8,458/7,715
Female	22,880/2,874/2,480	71,629/8,535/9,767	94,509/11,409/12,247
Total	24,267/3,056/2,660	138,503/16,811/17,302	162,770/19,867/19,962

Table 4. UTKFace dataset statistics.

(train/val/test)	Young (age ≤ 35)	Old (age > 35)	Total
Male	4,133/1,378/1,378	3,301/1,101/1,100	7,434/2,479/2,478
Female	4,931/1,643/1,644	1,858/619/619	6,789/2,262/2,263
Total	9,064/3,021/3,022	5,159/1,720/1,719	14,223/4,741/4,741

B.2. Metrics

Besides the weighted accuracy (WACC) and Area under the ROC Curve (AUC) (Fawcett, 2004), in terms of different fairness constraints tested in the experiment, we also use the following metrics to evaluate the model performance:

- Equalized Odds Difference (the smaller the better):
  $\max\{|\mathbb{P}(\hat{y} = 1 \mid a = 0, y = 0) - \mathbb{P}(\hat{y} = 1 \mid a = 1, y = 0)|, |\mathbb{P}(\hat{y} = 1 \mid a = 0, y = 1) - \mathbb{P}(\hat{y} = 1 \mid a = 1, y = 1)|\}$
- Accuracy Equality Difference (the smaller the better):
  $|\mathbb{P}(\hat{y} \neq y \mid a = 0) - \mathbb{P}(\hat{y} \neq y \mid a = 1)|$
- Worst Accuracy (the larger the better):
  $\min\{\mathbb{P}(\hat{y} = 0 \mid a = 0, y = 0), \mathbb{P}(\hat{y} = 0 \mid a = 1, y = 0), \mathbb{P}(\hat{y} = 1 \mid a = 0, y = 1), \mathbb{P}(\hat{y} = 1 \mid a = 1, y = 1)\}$

When report the experiment results, we use EO_Diff, AE_Diff, and WA to denote these metrics respectively. Furthermore, for the final evaluation criterion, we place equal weight on model prediction accuracy and fairness. To be more formal, we introduce an additional metric that combines both aspects - the **AF** metric. This metric is defined as an equal-weight linear combination of weighted accuracy and fairness metric, with both receiving equal weighting. Specifically, for Equalized Odds, AF = WACC - EO_Diff; for Accuracy Equality, AF = WACC - AE_Diff; while for Max-Min Fairness, AF = WACC + WA. We use different signs for different fair metrics, so that a larger AF value always indicates better performance of the model. We run all the experiments with twenty random seeds and report the mean over different trials.

B.3. Hyper-parameters

The hyper-parameters shared among all methods are tuned using the search ranges shown in the Table 5. For all the four last layer methods, we use full-batch SGD to train the model. We select the values of the hyper-parameters that lead to the highest AF value for each method, and list them in Table 6 and Table 7 for CelebA and UTKFace, respectively.

Table 5. Hyper-parameter search ranges.

Hyper-parameter	**search range**
learning rate	$[3 \times 10^{-4}, 1 \times 10^{-3}, 3 \times 10^{-3}]$
batch size	Full
number of epochs	[500, 1000, 1500, 2000]
α (in Eq. 2&4)	[0.5, 1, 2, 5, 10]

Table 6. Hyper-parameter settings for CelebA.

Methods	learning rate	number of epochs	α
LastFT	1×10^{-3}	1000	/
LastFT-RW	3×10^{-3}	1000	/
LastFT-Reg (EO)	1×10^{-3}	500	10
FDR (EO)	1×10^{-3}	1000	2
LastFT-Reg (AE)	1×10^{-3}	1000	0.5
FDR (AE)	1×10^{-3}	500	5
LastFT-Reg (MMF)	1×10^{-3}	1000	/
FDR (MMF)	1×10^{-3}	1000	/

Table 7. Hyper-parameter settings for UTKFace.

Methods	learning rate	number of epochs	α
LastFT	1×10^{-3}	1000	/
LastFT-RW	3×10^{-3}	1000	/
LastFT-Reg (EO)	1×10^{-3}	1000	0.5
FDR (EO)	1×10^{-3}	1500	2
LastFT-Reg (AE)	1×10^{-3}	1000	1
FDR (AE)	3×10^{-3}	1500	5
LastFT-Reg (MMF)	1×10^{-3}	1000	/
FDR (MMF)	1×10^{-3}	1000	/

C. Additional Experiments

C.1. Assessing Fairness on UTKFace Dataset

We also conduct the experiments on the UTKFace dataset with the goal of accurately predicting the age of the person in each image. The results of three different fairness notions are presented in Table 8. Based on our experiments, we find that (1) FDR has the best fairness metrics and the best overall performance (indicated by the highest AF), across all three fairness metrics. This suggests that the proposed method is efficient in improving fairness of the deep neural network. (2) FDR and LastFT-Reg have relatively low test WACC compared to other methods, indicating that adding fairness constraints may have a negative impact on the model's prediction accuracy.

C.2. Assessing Fairness under Transfer Learning Setting

To show that fairness can still be satisfied despite a change in data distribution, we explore the task that involves adapting a pre-trained model on the large ImageNet dataset to the CelebA dataset for the purpose of hair-color prediction, and the

Table 8. Last layer fine-tuning results with fairness notions on UTKFace dataset. Mean are reported over twenty random trials. Notably, methods including LastFT and LastFT-RW have the same value for WACC and AUC across different fairness notions since they do not depend on them, while other methods have different scores for different fairness notions.

Fairness Notion 1: EO	WACC		AUC		EO_Diff		AF
	Train	Test	Train	Test	Train	Test	Test
FullFT-Reg	0.998	**0.804**	1.000	**0.892**	0.002	0.141	0.663
LastFT	0.821	0.793	0.902	0.880	0.126	0.152	0.641
LastFT-RW	0.864	0.797	0.938	0.878	0.069	0.104	0.693
LastFT-Reg	0.821	0.793	0.901	0.879	0.140	0.140	0.653
FDR	0.848	0.781	0.928	0.863	0.011	**0.026**	**0.755**
Fairness Notion 2: AE	WACC		AUC		AE_Diff		AF
	Train	Test	Train	Test	Train	Test	Test
FullFT-Reg	0.998	**0.805**	1.000	**0.892**	0.001	0.037	0.768
LastFT	0.821	0.793	0.902	0.880	0.027	0.029	0.764
LastFT-RW	0.864	0.797	0.938	0.878	0.012	0.026	0.771
LastFT-Reg	0.822	0.791	0.901	0.878	0.010	0.028	0.763
FDR	0.857	0.785	0.936	0.866	0.003	**0.011**	**0.774**
Fairness Notion 3: MMF	WACC		AUC		WA		AF
	Train	Test	Train	Test	Train	Test	Test
FullFT-Reg	0.999	**0.814**	1.000	**0.899**	0.997	0.710	1.524
LastFT	0.821	0.793	0.902	0.880	0.736	0.689	1.482
LastFT-RW	0.864	0.797	0.938	0.878	0.825	0.727	1.524
LastFT-Reg	0.807	0.779	0.892	0.866	0.776	0.726	1.505
FDR	0.851	0.788	0.930	0.870	0.829	**0.745**	**1.533**

UTKFace dataset for age prediction, using last layer fine-tuning. We select Equalization Odds (EO) to illustrate our idea. The results are presented in Table 9&10. To avoid bias, the FullFT-Reg method is excluded from this task. From the results, Equalized Odds can be effectively maintained during OOD-fine-tuning using FDR in both two datasets. Additionally, FDR also achieves the best overall performance (highest AF value) without hurting the WACC and AUC significantly.

C.3. Further Exploration with Surgical Fine-tuning

Lee et al. (2022) proposed a modification to the last layer fine-tuning by extending it to different blocks which consists of a set of layers of the model. For example, the ResNet-18 architecture can be divided into six blocks: an input layer which is an initial convolutional layer; Blocks 1, 2, 3 and 4, each of which comprised of multiple convolutional layers with batch normalization and activation functions, followed by a shortcut connection; and the last layer. During the fine-tuning, they select one block to update and fix the parameters of other layers. They also propose several criteria for determining the appropriate subset of layers to perform fine-tuning, such as Auto-RGN (Lee et al., 2022) which automatically selects an appropriate subset of layers for fine-tuning.

Inspired by that, we conduct the similar experiments in the fairness setting. In our experiments (results are shown in Table 11), different blocks of the model are fine-tuned on the balanced sampled dataset while incorporating fairness constraints. According to the results, fine-tuning any set of layers other than the input layer can effectively address the fairness over-fitting issue without affecting the ACC/AUC performance of the model. Fine-tuning Block 1 typically performs better than fine-tuning other blocks. This finding suggests that it is promising to explore more sophisticated fine-tuning strategies, which opens an interesting direction for future work.

Table 9. Last layer fine-tuning results with fairness notions in the transfer learning setting on **CelebA** dataset. For WACC, AUC and AF, a larger value is considered better; while for EO_Diff, a smaller value is considered better. Mean are reported over twenty random trials.

Fairness Notion: Equalized Odds (smaller value is better)							
	WACC		AUC		EO_Diff		AF
	Train	Test	Train	Test	Train	Test	Test
LastFT	0.899	**0.871**	0.961	**0.944**	0.252	0.412	0.459
LastFT-RW	0.871	0.856	0.943	0.930	0.070	0.114	0.742
LastFT-Reg	0.893	0.867	0.959	0.941	0.168	0.302	0.565
FDR	0.854	0.841	0.925	0.915	0.021	**0.062**	**0.779**

Table 10. Last layer fine-tuning results with fairness notions in the transfer learning setting on **UTKFace** dataset. For WACC, AUC and AF, a larger value is considered better; while for EO_Diff, a smaller value is considered better. Mean are reported over twenty random trials.

Fairness Notion: Equalized Odds (smaller value is better)							
	WACC		AUC		EO_Diff		AF
	Train	Test	Train	Test	Train	Test	Test
LastFT	0.751	0.695	0.833	0.766	0.174	0.185	0.510
LastFT-RW	0.714	**0.702**	0.790	**0.775**	0.113	0.127	0.575
LastFT-Reg	0.729	0.687	0.817	0.761	0.214	0.197	0.490
FDR	0.695	0.682	0.768	0.752	0.011	**0.033**	**0.649**

Table 11. Surgical fine-tuning results with Equalized Odds fairness notion on balanced-sampled CelebA dataset. Mean are reported over twenty random trials.

Metrics	WACC			AUC			Fairness Metric		
Fairness Constraint	EO	AE	MMF	EO	AE	MMF	EO	AE	MMF
Last Layer	0.863	0.877	0.882	0.935	0.963	0.962	0.109	0.009	0.781
Input Layer	0.916	0.919	0.918	0.948	0.947	0.947	0.182	0.062	0.423
Block 1	0.903	0.906	0.886	0.963	0.963	0.962	**0.071**	0.010	**0.860**
Block 2	0.896	0.907	0.890	0.965	0.964	0.964	0.081	0.011	0.833
Block 3	0.883	0.903	0.895	0.959	0.966	0.964	0.084	0.006	0.812
Block 4	0.883	0.890	0.895	0.957	0.961	0.961	0.135	0.010	0.770
Auto-RGN	0.888	0.895	0.895	0.963	0.966	0.965	0.096	**0.005**	0.778