Robust and Personalized Federated Learning with Spurious Features: an Adversarial Approach

Xiaoyang Wang[*] **Han Zhao, Klara Nahrstedt, Sanmi Koyejo**
Department of Computer Science
University of Illinois at Urbana–Champaign
{xw28, hanzhao, klara, sanmi}@illinois.edu

Abstract

The most common approach for personalized federated learning is fine-tuning the global machine learning model to each client. While this addresses some issues of statistical diversity, we find that such personalization methods are vulnerable to *spurious features*, leading to bias and sacrificing generalization. Nevertheless, debiasing the personalized models is difficult. To this end, we propose a strategy to mitigate the effect of spurious features based on an observation that the global model in the federated learning step has a low bias degree due to statistical diversity. Then, we estimate and mitigate the *bias degree difference* between the personalized and global models using *adversarial transferability* in the personalization step. Empirical results on MNIST, CelebA, and Coil20 datasets show that our method improves the accuracy of the personalized model on the bias-conflicting data samples by up to 14.3%, compared to existing personalization approaches, while preserving the benefit of enhanced average accuracy from fine-tuning.

1 Introduction

Recent works [Fallah et al., 2020, T. Dinh et al., 2020, Li et al., 2021] on personalized federated learning (FL) fine-tune the global model on each client's local dataset. Although theoretical and empirical results show that the personalized models fit the local data better and improve the accuracy, few works consider what features the personalized models learn from the local dataset. Our motivating hypothesis is that: *Not all features are beneficial for a machine learning (ML) model*. For example, in a gender prediction task using face images, the ML model could learn to predict gender based on hair color because females are more likely to have blond hair [Sagawa et al., 2019]. However, the hair color is a *spurious feature* because it only statistically correlates with the gender label but does not imply causation. The accuracy of an ML model that relies on spurious features such as hair color could drop significantly on the data where the spurious correlation does not hold, e.g., the blond male images [Sagawa et al., 2019]. Such an accuracy disparity caused by spurious features leads to issues in both fairness (racial bias) [Khani and Liang, 2021] and robustness (accuracy decrease under distribution shift) [Koh et al., 2021]. An ML model has a high *bias degree* if the accuracy disparity is large. Compared to the global model, *the personalized models are more vulnerable to spurious features and therefore have a higher bias degree*.

Empirically, we observe that the non-i.i.d. nature of the data distribution in the FL setting reduces the bias degree of the global model. The reason is that the distribution shift of spurious features across users could be larger than that of non-spurious features, making learning spurious features difficult, as Figures 1 and 2 show. In Section 2, we will show that the global model is more robust against both the synthetic spurious feature in MNIST and Coil20 datasets and the real spurious feature in the CelebA dataset, with explanations. However, the personalization methods fine-tune the global model

[*] Corresponding author.

1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL 2021), Virtual Meeting.

locally, leading to a biased personalized model. Nevertheless, *debiasing the personalized models remains a challenge*.

Various methods have been developed to disentangle spurious features from ML models, but few apply to the personalized models. Firstly, many prior works [Li and Vasconcelos, 2019, Sagawa* et al., 2020] rely on human supervision. For example, the group distributional robust optimization (DRO) method [Sagawa* et al., 2020] aims to balance the error rate of ML models across different manually annotated groups, which increases the labeling cost. Additionally, the group DRO method is sensitive to imprecise group annotations [Liu et al., 2021]. For the methods that do not require human supervision, access to bias-conflicting samples (e.g., blond male images in CelebA dataset) is necessary. Residual learning-based methods [He et al., 2019, Nam et al., 2020, Liu et al., 2021] train a biased ML model and up-weight the residual, which mainly contains bias-conflicting samples that the biased ML model mis-predicts. Other methods introduce auxiliary neural networks to extract superficial statistics (e.g., texture bias) and enforce independence between the prediction and the superficial statistics, which also needs bias-conflicting samples [Wang et al., 2019]. However, the bias-conflicting samples are rare. In the CelebA dataset, only around 1.7k samples are bias-conflicting out of more than 170k samples. If we distribute the CelebA dataset across users, nearly half do not have any bias-conflicting sample. Prior work [Li and Wang, 2019] has explored creating a global proxy dataset for training FL models. However, a global proxy dataset may not characterize the local samples well. Besides the challenge in model debiasing, *estimating the bias degree of the personalized model becomes difficult without access to bias-conflicting samples*.

To address the supervision and data limitations, we developed an intuitive and effective method to reduce the bias degree of the personalized models without relying on bias-conflicting samples. Inspired by prior works [Tramèr et al., 2017, Liang et al., 2021] on the transferability of adversarial examples, we use the *adversarial transferability* between the low bias degree global model and the personalized models as a proxy to estimate the bias degree of the personalized models. The idea is that if two ML models use disjoint subsets of features, the adversarial examples that one ML model generates can not transfer to the other ML model. For example, to attack an unbiased ML model trained on the CelebA dataset, the adversary needs to add perturbations to the non-spurious shape features. However, an image with shape perturbations can not attack a biased ML model, which uses the spurious color feature to predict. Empirically, we show that the adversarial transferability between the global and personalized models strongly correlates with the bias degree of the personalized models in Section 2. Based on the observed correlation, we develop a method that enforces the adversarial transferability between the global and the personalized models in Section 3. However, we find that only enforcing adversarial transferability is insufficient because the personalized model may forget the rare bias-conflicting samples. The forgetting also increases the accuracy disparity. To address the forgetting issue, we developed another method based on loss function approximation. Empirical results show that combining the two methods reduces the bias degree of personalized models. We also include experiments highlighting the benefits of the approach, showing that it results in 3.85% average accuracy improvement on the biased test set and 0.8% accuracy improvement on the biased-conflicting test set compared to the global model. In contrast, naive fine-tuning decreases the accuracy on biased-conflicting test set by up to 14.3% on MNIST, CelebA and Coil20 datasets. We theoretically connects the adversarial transferability and the bias degree difference between the global and personalized models, which is presented in Appendix **??** due to the limited space.

Our contributions are summarized as follows:

- We empirically evaluated the bias degree of the global and personalized models in a federated learning setting with spurious features, highlighting the risk of existing methods.
- We designed a method to estimate the bias degree of the personalized models, based on the low bias degree global model and the adversarial transferability between the global and personalized models.
- We developed two methods to reduce the bias degree of personalized models by (1) enforcing the adversarial transferability between the global and personalized models and (2) preventing the personalized model from forgetting bias-conflicting samples, which are rare or missing in the local dataset.
- We theoretically connected the adversarial transferability and the bias degree difference of the global and personalized models.
- We conduct extensive experiments, validating the proposed method and showing the benefits.

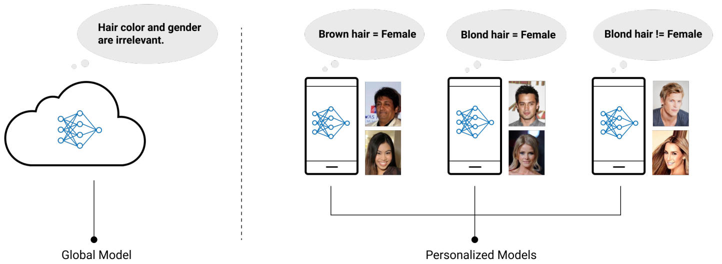


Figure 1: Spurious Correlation Varies across Users. The variation results in a global machine learning model with lower bias degree in a federated learning setting.

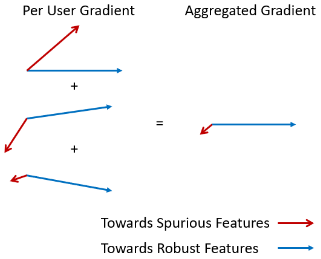
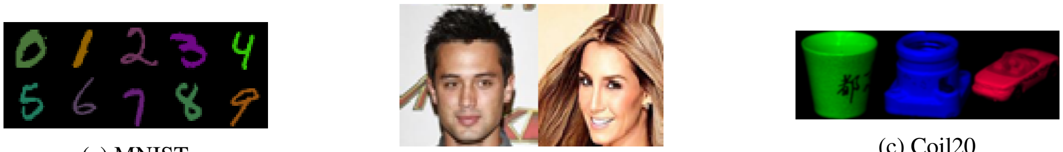


Figure 2: Gradient Divergence between Users. The divergence slows down the learning of spurious features.



(a) MNIST

(b) CelebA

(c) Coil20

Figure 3: Datasets with Spurious Features. The object color spuriously correlates with the label in MNIST and Coil20 datasets. The hair color spuriously correlates with gender in the CelebA dataset.

2 An Empirical Study on Personalized Federated Learning with Spurious Features

This section presents our empirical study on the bias degree of the global and personalized models in an FL setting. The results highlight the risk of existing personalization methods and the difficulty of mitigating the risk. Additional results show the correlation between the adversarial transferability and the bias degree difference between the global and personalized models. The observed correlation is the basis for our debiasing method for the personalized models, presented in Section 3.

2.1 Background and Setup

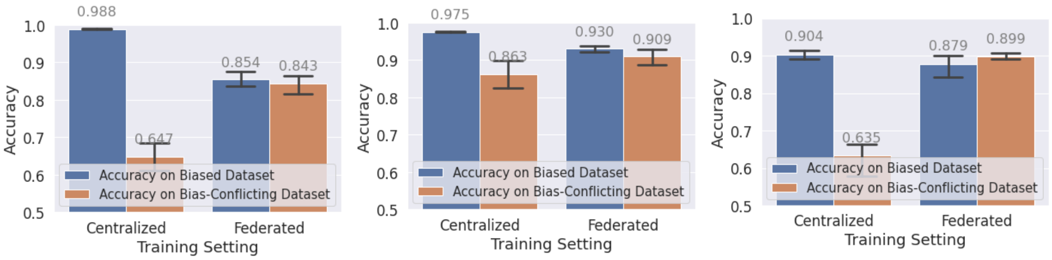
We briefly introduce the background and setup of our empirical study. A more detailed description of the experimental setting is in Section 4.

Spurious Features We consider color as the spurious feature for the MNIST, CelebA, and Coil20 datasets. In the MNIST and Coil20 datasets, we manually color the objects according to their labels to create spurious correlations, as Figure 3 shows. In the CelebA dataset, the hair color attribute correlates with the gender label.

Bias-aligned and Bias-conflicting Samples Given a biased ML model that predicts using spurious features, the bias-aligned samples are the samples on which the biased ML model makes correct predictions [Nam et al., 2020]. The rest of the samples are bias-conflicting. For example, in CelebA dataset, blond female images are bias-aligned, and blond male images are bias-conflicting.

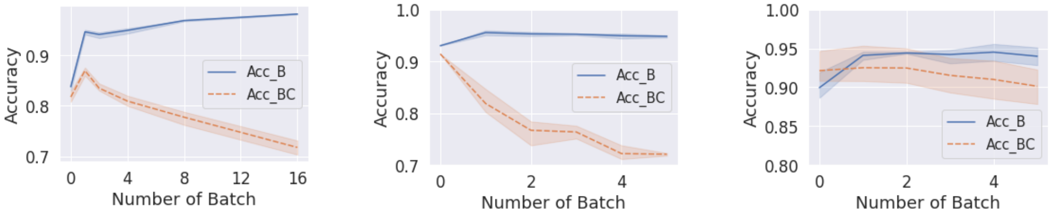
Data Partition In the federated setting, each user has data from 5 different classes in the MNIST and Coil20 dataset and 20 celebrities in the CelebA dataset. The spurious correlations vary across clients for the MNIST and Coil20 data (e.g., the red color correlates with label zero on the first client and with label one on the second client) to create additional statistical diversity. In the centralized setting, we use the same dataset with fixed spurious correlations. We create a biased test set, which follows the same data distribution as the train set, and a bias-conflicting test set, which only contains bias-conflicting samples.

Training Method We train the global model using federated averaging algorithm [McMahan et al., 2017], which learns a model $f : \mathcal{X} \to \mathcal{Y}$ that minimizes: $\mathcal{L}(f, \mathcal{D}_g) = \sum_{i=1}^{N} \frac{|\mathcal{D}_{l_i}|}{|\mathcal{D}_g|} \cdot \mathbb{E}_{\boldsymbol{x},y \sim \mathcal{D}_{l_i}} \ell(f(\boldsymbol{x}), y)$, where N is the number of clients, $\mathcal{D}_{l_i}$ is the local dataset for client i, $\mathcal{D}_g = \bigcup_{i=1}^{N} \mathcal{D}_{l_i}$ is the global dataset, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is the loss function.



(a) MNIST (b) CelebA (c) Coil20

Figure 4: The Accuracy of ML Models on Biased Dataset and Bias-Conflicting Dataset with Different Training Settings. The global models in the federated setting achieve smaller accuracy disparities between biased and bias-conflicting dataset.



(a) MNIST (b) CelebA (c) Coil20

Figure 5: The Accuracy of Personlized Model on Biased Dataset and Bias-Conflicting Dataset with Different Number of Fine-tuning Batches. The personalized models entangle spurious features and increase accuracy disparities between biased and bias-conflicting dataset.

Adversarial Examples We could generate an adversarial example $\boldsymbol{x}_{adv}$ using data sample $\boldsymbol{x}$ with label y by solving:

$$\boldsymbol{x}_{adv} = \mathop{\arg\max}_{||\boldsymbol{x}_{adv}-\boldsymbol{x}||\le\epsilon} \ell(f(\boldsymbol{x}_{adv}), y) \tag{1}$$

, where f is the victim ML model, ℓ is the loss function and ϵ is the attack budget. An adversarial example is transferable if it fools another ML model (e.g., a personalized model) other than the original victim model f (e.g., the global model).

2.2 Statistical Diversity Reduces the Bias Degree

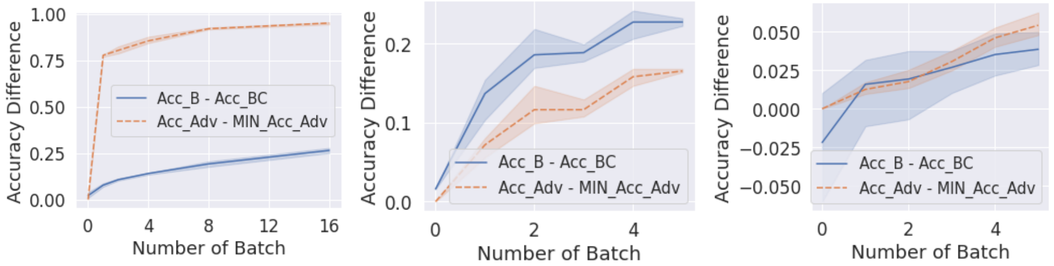
Figure 4 shows the accuracy disparities of ML models on bias and bias-conflicting test sets, which indicates their bias degree. Compared to the models trained in the centralized setting, the bias degree of models trained in the federated setting decreased significantly. These empirical results suggest that the global model in FL is more robust to spurious features if the spurious features are non-i.i.d. across clients.

Intuitive Explanation Let us consider the relationship between the gradient directions and the learned features. For the spurious features, its correlation with the label may change across clients. As an example, in some users' local datasets, the blond hair does not correlate with the gender label because the dataset does not contain any blond female image or the dataset has blond male images, as is visualized in Figure 1. Therefore, the gradient directions for the spurious features could diverge across clients, as is visualized in Figure 2. The divergence between the gradient directions, as a consequence, makes learning spurious features difficult. In contrast, the non-spurious features, e.g., shape features, are more consistent across clients, leading to a more consistent gradient direction.

2.3 Personlization Increases the Bias Degree

Although the global model in the federated setting has a lower bias degree than that in the centralized setting, the advantage could vanish during the personalization step. Figure 5 shows the accuracy disparity of the personalized model during personalization, from which we could observe:

Personalized Model Entangles Spurious Features The accuracy first increases and decreases on the MNIST bias-conflicting test set and slowly decreases on that of Coil20. These two observations



(a) MNIST

(b) CelebA

(c) Coil20

Figure 6: The Accuracy Disparity of Personalized Models on Biased Dataset and Bias-Conflicting Dataset (Acc_B - Acc_BC) and their Accuracy Increase on Adversarial Examples (Acc_Adv - MIN_Acc_Adv). As the personalized models entangle spurious features and increase the accuracy disparity, the accuracy of the personalized models on adversarial examples increases, which indicates the adversarial transferability between the global and personalized models decreases. MIN_Acc_Adv is 0 because the adversarial examples transfer to the personalized models with no update.

indicate that the personalized model needs a few batches to learn spurious features. Therefore, one may wonder if early-stopping would help. However, since the accuracy drop on the bias-conflicting dataset may not be observable, as is discussed in Section 1, one has no clue on when to stop, and the observable accuracy on the biased test set would mislead to biased models.

2.4 Adversarial Transferability Indicates Bias Degree Difference

Liang et al. [2021] uncovers the connection between adversarial transferability and knowledge transferability, which motivates our method. To estimate the bias degree of the personalized models, which is considered difficult without bias-conflicting samples, we use the transferability of adversarial examples between the global and personalized models as a proxy. Adversarial transferability is defined using the accuracy of personalized models on the adversarial examples generated by the global model—the *higher* the accuracy, the *lower* the transferability. Figure 6 plots the accuracy disparity and the adversarial transferability during fine-tuning. As the accuracy disparity increases, which indicates the bias degree of personalized models increases, the accuracy on adversarial examples also increases, which indicates the adversarial transferability decreases.

3 Methods

Based on the observation in Section 2.4, we first introduce transferable adversarial examples to the personalization step, aiming to enforce the adversarial transferability. However, the accuracy disparity still increases, albeit much slower, even if the adversarial transferability remains high. One possible reason is that the personalized model forgets the bias-conflicting samples, which are rare or missing. Therefore, we introduce another method based on loss function approximation to mitigate forgetting. Combining the two proposed methods, we address the accuracy disparity. We note that both methods are lightweight, which fit the resource constraint client devices in FL systems.

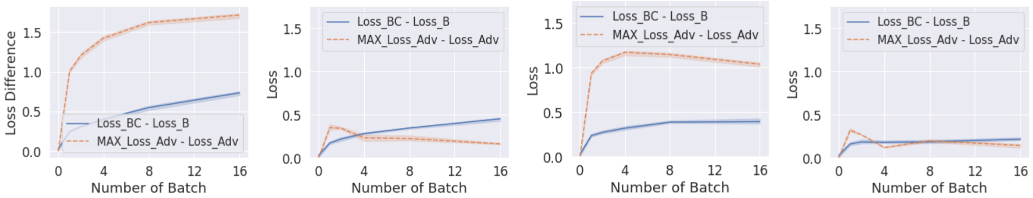
3.1 Training Adversarial Transferability

We enforce the global and personalized models to make consistent predictions on the adversarial examples, such that the adversarial examples could transfer from one to another.

Generating Adversarial Examples Projected gradient descent (PGD) attack [Madry et al., 2018] is among the most effective attacks that utilize the neural network's first-order gradient, which is easy to compute. The attack solves $\boldsymbol{x}_{adv} = \arg\max_{||\boldsymbol{x}_{adv}-\boldsymbol{x}||\leq\epsilon} \ell(f(\boldsymbol{x}_{adv}), y)$ iteratively. At iteration $t+1$, the adversarial example is: $\boldsymbol{x}_{adv}^{t+1} = \mathrm{Proj}_{||\boldsymbol{x}_{adv}-\boldsymbol{x}||\leq\epsilon}(\boldsymbol{x} + \alpha \cdot \mathrm{sign}(\nabla_{\boldsymbol{x}_{adv}^{t}} \ell(f_g(\boldsymbol{x}_{adv}^{t}), y)))$

where $f_g : \mathcal{X} \to \mathcal{Y}$ is the global model, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is the loss function, and Proj is a projection operator.

Enforcing Consistent Prediction Both the global model f_g and the personalized model f_p take the adversarial example $\boldsymbol{x}_{adv}$ as input and output $\boldsymbol{z}_g$ and $\boldsymbol{z}_p$ from their last layers, respectively. We enforce the adversarial transferability by adding the following regularization term, which maximizes



(a) Naive Fine-tuning (b) Training Adversarial Transferability (c) Mitigating Forgetting (d) Applying Both Methods

Figure 7: The Loss Disparity and Adversarial Transferability with Different Methods. Combining the two proposed methods flats the loss disparity.

the cross-entropy between $\boldsymbol{z}_g$ and $\boldsymbol{z}_p$. Since the global model f_g is fixed as a reference in the personalization step and its low accuracy disparity is desirable, we use $\boldsymbol{z}_g$ as the ground-truth:

$$R_{adv}(\boldsymbol{z}_g, \boldsymbol{z}_p) = \sum_{i=1}^{K} [z_{g_i} \cdot \log(z_{p_i}) + (1 - z_{g_i}) \cdot \log(1 - z_{p_i})] \tag{2}$$

where K is the number of classes. The local model has access to the global model, so there is no additional communication overhead for implementing this regularization.

3.2 Mitigating Forgetting

Catastrophic-forgetting refers to the phenomenon that a neural network forgets the previous task while learning a new task. In our scenario, we suspect that the personalized model forgets the bias-conflicting samples. Therefore, we developed a method to address the forgetting issue, motivated by the recent advance in continual learning [Yin et al., 2020]. The idea is first approximating the loss function $\mathcal{L}(f_p, \mathcal{D}_g)$ of the personalized model f_p on the global dataset $\mathcal{D}_g$ that contains more bias-conflicting samples. Then, we could penalize the personalized model if the approximated loss increases, which is defined as $\mathcal{L}(f_p, \mathcal{D}_g) - \hat{\mathcal{L}}(f_g, \mathcal{D}_g)$.

Loss Function Approximation Direct estimation of the loss $\mathcal{L}(f_p, \mathcal{D}_g)$ is intractable because we do not have access to $\mathcal{D}_g$ on the client. Therefore, we apply a second-order Taylor expansion at the global model f_g with parameter $\boldsymbol{w}_g$ to approximate the loss function:

$$\begin{aligned}\mathcal{L}(f_p, \mathcal{D}_g) &= \mathcal{L}(f_g, \mathcal{D}_g) + (\boldsymbol{w}_p - \boldsymbol{w}_g)^\top \nabla_{\boldsymbol{w}} \mathcal{L}(f_g, \mathcal{D}_g) + \frac{1}{2}(\boldsymbol{w}_p - \boldsymbol{w}_g)^\top \nabla^2_{\tilde{\boldsymbol{w}}} \mathcal{L}(f_g, \mathcal{D}_g)(\boldsymbol{w}_p - \boldsymbol{w}_g) \\ &\leq \mathcal{L}(f_g, \mathcal{D}_g) + (\boldsymbol{w}_p - \boldsymbol{w}_g)^\top \nabla_{\boldsymbol{w}} \mathcal{L}(f_g, \mathcal{D}_g) + \frac{\lambda}{2} \cdot \|\boldsymbol{w}_p - \boldsymbol{w}_g\|^2\end{aligned}$$

where $\boldsymbol{w}_g$ and $\boldsymbol{w}_p$ are the weights for global and personalized model, respectively, and $\tilde{\boldsymbol{w}}$ is a linear interpolation between these two, by the Lagrange's mean-value theorem. Since we know that the global model converges to some minima, by the first-order optimality condition, we have $\nabla_{\boldsymbol{w}} \mathcal{L}(f_g, \mathcal{D}_g) = \boldsymbol{0}$. Furthermore, by the smoothness assumption on the loss function, we know the spectral norm of the Hessian, $\|\nabla^2_{\tilde{\boldsymbol{w}}} \mathcal{L}(f_g, \mathcal{D}_g)\|$, is upper bounded by λ [Nesterov, 2003, Proof of Theorem 2.1.5]. Combining all the above arguments, we have

$$\mathcal{L}(f_p, \mathcal{D}_g) - \mathcal{L}(f_g, \mathcal{D}_g) \leq \frac{\lambda}{2} \cdot \|\boldsymbol{w}_g - \boldsymbol{w}_p\|^2. \tag{3}$$

The upper bound in Eq. (3) implies that, by adding a simple L_2 regularization term $R_{L_2} = \lambda \cdot \|\boldsymbol{w}_g - \boldsymbol{w}_p\|^2$, we can mitigate the forgetting issue. Although prior works [Li et al., 2020, T. Dinh et al., 2020, Li et al., 2021] have explored similar regularization methods, we derive the regularization term from a different perspective.

4 Experiments

This section presents our experimental results, in addition to the results in Sections 2 and 3, demonstrating that our method prevents the personalized model from using spurious features and reduces their bias. We also show that the benefit of enhanced average accuracy from fine-tuning is preserved. Compared to a supervised up-weighting strategy from prior work [Sagawa et al., 2019], we find our method performs better, possibly because the number of bias-conflicting samples is small.

Table 1: Accuracy

Method	MNIST		CelebA_S		CelebA_R		Coil20	
	Acc_B	Acc_BC	Acc_B	Acc_BC	Acc_B	Acc_BC	Acc_B	Acc_BC
Global	.852 ± 2e-4	.847 ± 6e-4	.930 ± 5e-5	.910 ± 3e-4	.909 ± 6e-5	.929 ± 5e-5	.882 ± .6e-4	.903 ± 7e-4
FT	.989 ± 6e-7	.704 ± 3e-4	.952 ± 6e-5	.786 ± 5e-4	.963 ± 6e-6	.849 ± 1e-3	.958 ± 2e-5	.782 ± 8e-4
UW	.968 ± 2e-5	.823 ± 7e-4	.930 ± 7e-6	.889 ± 5e-4	.936 ± 1e-5	.895 ± 4e-4	N/A	N/A
Ours	.951 ± 2e-5	.870 ± 8e-4	.932 ± 3e-5	.910 ± 2e-4	.925 ± 2e-5	.927 ± 8e-5	.901 ± 3e-4	.916 ± 5e-8

[*] Accuracy on Biased Dataset / Accuracy on Bias-conflicting Dataset

4.1 Setting

We list more details about the dataset partition, hyper-parameters, and model selection in addition to Section 2.1.

Data Partition We distribute the MNIST and Coil20 dataset across 50 clients. Each client has data from 5 different classes. The local dataset on each client is further partitioned to train/validation/test set with a ratio of 72:8:20, following prior work [Li et al., 2021]. The test set here is biased. We alternate the spurious features in biased test sets by recoloring the data to create a bias-conflicting test set. The CelebA dataset is distributed among 508 clients according to the celebrity identity, following LEAF [Caldas et al., 2018]. Due to the limited space, we report the details for the synthetic CelebA partition CelebA_S and the real partition CelebA_R in Appendix A.

Spurious Features For the biased train/validation/test sets, we correlate 98% of data samples with spurious features in MNIST and CelebA_S and 100% in Coil20, similar to prior work [Nam et al., 2020]. The details of CelebA_R are in Appendix A.

Hyper-parameters We use Adam optimizer [Kingma and Ba, 2015] through our the experiments with learning rate $1e-4$. Although stochastic gradient descent (SGD) optimizer is more common in vision-related tasks, we find that the Adam optimizer always finds a less biased ML model in all the cases. We train the global model for 500 rounds. 5 clients are selected per round, and each performs 5 epochs of local updates. We tune the weight of adversarial transferability term and loss approximating term from $\{0.01, 0.1, 1.0, 10.0\}$ and select the largest value that does not decrease the validation accuracy during penalization.

Model Selection We select models using the validation accuracy minus the decrease of adversarial transferability. For other baseline and competitor methods, we use the validation accuracy for model selection.

4.2 Main Result

We compare our method to no personalization (Global), naive fine-tuning (FT), and up-weighting (UW) [Sagawa et al., 2019]. The up-weighting method is implemented via sampling bias-aligned and bias-conflicting samples with equal probability [Sagawa et al., 2019]. Each experiment repeats 9 times with 3 random seeds for the federated learning step and 3 for the personalization step.

Tables 1 shows the main result. Our method results in $\mathbf{3.85}\%$ average accuracy improvement on the biased test set and $\mathbf{0.8}\%$ accuracy improvement on the biased-conflicting test set compared to the global model, on average. In contrast, the naive fine-tuning method sacrifices the accuracy on the bias-conflicting test set by up to 14.3%. We also find that our methods outperform the supervised up-weighting method, which decreases the accuracy on the biased-conflicting test set by 2.63% on average. One possible reason is that the diversity of the up-weighted bias-conflicting samples are small. Therefore, the neural network could memorize them instead of discarding spurious features. Appendix B details the analysis on the impact of the diversity of bias-conflicting samples on debiasing, showing that our method is applicable in the scarcity of bias-conflicting samples while the up-weighting method fails.

5 Conclusion

In this work, we explored the risk of federated learning personalization methods in the presents of spurious features, which lead to biased personalized models. To mitigate the risk, we developed a strategy based on the adversarial transferability between the global and personalized models. Empirical results show that enforcing transferability reduces the bias degree of personalized models.

References

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33, 2020.

Canh T. Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21394–21405. Curran Associates, Inc., 2020.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Fereshte Khani and Percy Liang. Removing spurious features can hurt accuracy and affect groups disproportionately. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 196–205, 2021.

Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9572–9581, 2019.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=ryxGuJrFvS`.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. *arXiv preprint arXiv:1908.10763*, 2019.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. In *Advances in Neural Information Processing Systems*, 2020.

Haohan Wang, Zexue He, Zachary L. Lipton, and Eric P. Xing. Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rJEjjoR9K7`.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

Kaizhao Liang, Jacky Y Zhang, Boxin Wang, Zhuolin Yang, Sanmi Koyejo, and Bo Li. Uncovering the connections between adversarial transferability and knowledge transferability. In *International Conference on Machine Learning*, pages 6577–6587. PMLR, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJzIBfZAb`.

Dong Yin, Mehrdad Farajtabar, Ang Li, Nir Levine, and Alex Mott. Optimization and generalization of regularization-based continual learning: a loss approximation viewpoint. *arXiv preprint arXiv:2006.10974*, 2020.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020. URL `https://proceedings.mlsys.org/paper/2020/file/38af86134b65d0f10fe33d30dd76442e-Paper.pdf`.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

A More Details on Data Partition

For the CelebA dataset, we consider two partitions. In the first partition, each client represents 20 celebrities. One celebrity only appears on one client. We have 508 clients in total. The blond male images in the biased test sets are copied to bias-conflicting test sets. We use all clients for training the global model. In the personalization step, we select the clients who have more than five blond female training samples and more than five blond male test samples. We select these clients because they have enough samples to create spurious correlations and bias-conflicting test sets. Although the first partition on CelebA is real, the number (161) of blond male images is small. To make the result more convincing, we create another synthetic CelebA partition.

In the synthetic CelebA partition, there are 650 blond male images, which achieve a similar bias-conflicting test set size as prior works Sagawa et al. [2019], Liu et al. [2021]. The 650 images are distributed to 3 clients with 2350 other images. The rest of the images are distributed in the same way as the first partition. Tables 2, 3, and 4 provide more details about the 3 clients.

Table 2: Number of Train and Validation Samples in CelebA_S

Client ID	Non-blond Female	Non-Blond Male	Blond Female	Blond Male
0	55	31	12	2
1	30	68	0	2
2	59	28	11	2

Table 3: Number of Biased Test Samples in CelebA_S

Client ID	Non-blond Female	Non-Blond Male	Blond Female	Blond Male
0	115	60	45	2
1	60	75	79	0
2	86	111	14	0

Table 4: Number of Biased-Conflicting Test Samples in CelebA_S

Client ID	Non-blond Female	Non-Blond Male	Blond Female	Blond Male
0	0	0	0	200
1	0	0	0	203
2	0	0	0	204

B The Diversity of Bias-conflicting Samples Impacts Debiasing

To explore the impact of the diversity of bias-conflicting samples on debiasing, we vary the diversity of bias-conflicting samples and adjust the up-weighting factors accordingly. Specifically, we sample a factor of 0.02, 0.025, 0.033, 0.05, and 0.1 biased data samples from the MNIST dataset and re-color them to become bias-conflicting. The factor in the sampling step is called sampling factor. Then, we up-weight the bias-conflicting samples by a factor of 50, 40, 30, 20, 10, respectively, keeping the total number of bias-conflicting samples consistent. Here, the bias-conflicting samples have less diversity if generated by a small number of biased data samples with a large up-weighting factor. Experimental results in Figure 8 show that, as the diversity reduces, the accuracy disparity of personalized model on the biased dataset and bias-conflicting dataset increases, supporting our analysis. Therefore, our method is applicable in the scarcity of bias-conflicting samples while the up-weighting method fails.

C Theoretical Insights

This section presents our theoretical result that supports our hypothesis in Section 1 and the experimental results in Section 2.4. Our theoretical result applies the loss, which has similar behavior to

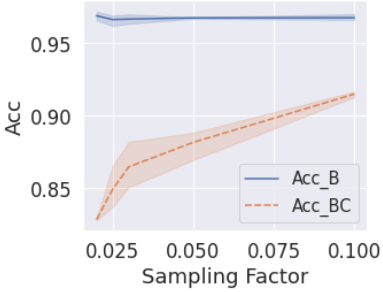


Figure 8: The up-weighting method is less effective, resulting in large accuracy disparity of personalized model on biased dataset and bias-conflicting dataset, if the bias-conflicting samples is generated by a small number of biased data samples using a small sampling factor and a large up-weighting factor. The up-weighting factor is set to be the reciprocal of the sampling factor.

Table 5: Table of Notation

Symbol	**Description**
$\boldsymbol{x}, y$	A pair of data sample and label
$\boldsymbol{x}_r, \boldsymbol{x}_s$	The robust feature and spurious features in $\boldsymbol{x} = [\boldsymbol{x}_r, \boldsymbol{x}_s]$, respectively
d, d_r, d_s	The dimension of $\boldsymbol{x}$, $\boldsymbol{x}_r$, $\boldsymbol{x}_s$, respectively
f_g	The global model
f_p	The personlized local model
$\boldsymbol{\delta}_{f_g,\epsilon}$	An adversarial purtubation generated by the global model f_g with attack budget ϵ
$\boldsymbol{\Delta}$	An natural perturbation, which could flip the spurious attribute
$\mathcal{D}_g$	The global distribution, which is the union of local distributions
$\mathcal{D}_b$	A biased local distribution
$\mathcal{D}_{bc}$	A bias-conflicting local distribution
$\mathcal{D}_{\boldsymbol{\Delta}\|\boldsymbol{x},y}$	The distribution of natural perturbation
$\text{supp}(\mathcal{D})$	The support of distribution $\mathcal{D}$
$\langle\cdot,\cdot\rangle$	An inner product of two vectors
$\cdot \frown \cdot$	A concatenation of two vectors

the accuracy as is shown in the empirical results in Section 4. Before we proceed, some additional definitions and notations are needed for the presentation, and we provide Table 5 summarizing all the notations used to ease the reading. Then, we connect both the loss disparity and the adversarial transferability to the angle between the gradients of the global and personalized models. In what follows, we shall show an upper bound of the loss disparity of a personalized model, which consists of the adversarial transferability between the global and personalized models and an indicator of the entanglement of the global model to spurious features.

C.1 More Definitions and Notation

We define natural perturbation $\boldsymbol{\Delta}$ to model the distribution shift between the bias-conflicting $\mathcal{D}_{bc}$ and biased $\mathcal{D}_b$. $\boldsymbol{\Delta}$ could change a bias-aligned sample to a corresponding bias-conflicting sample. The distribution $\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}$ of the natural perturbation $\boldsymbol{\Delta}$ conditions on the data sample $\boldsymbol{x}$. Formally, for any $\boldsymbol{x} \in \mathcal{R}^{\mathrm{d}}$, we have: $\Pr_{\boldsymbol{x}\sim\mathcal{D}_{bc}}(\boldsymbol{x}) = \sum_{\boldsymbol{x}'\in\mathcal{R}^{\mathrm{d}},\boldsymbol{\Delta}\in\mathcal{R}^{\mathrm{d}}} \mathbf{1}_{\{\boldsymbol{x}=\boldsymbol{x}'+\boldsymbol{\Delta}\}} \cdot \Pr_{\boldsymbol{x}'\sim\mathcal{D}_b}(\boldsymbol{x}') \cdot \Pr_{\boldsymbol{\Delta}\sim\mathcal{D}_{\Delta}|\boldsymbol{x}'}(\boldsymbol{\Delta})$.

Running Example For a non-blond male image $\boldsymbol{x}$, we could draw a natural perturbation $\boldsymbol{\Delta}$ from $\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}$ that change the hair color in $\boldsymbol{x}$ to blond. That is saying, $\boldsymbol{x}+\boldsymbol{\Delta}$ is a blond male image. Iteratively drawing data samples from the biased dataset and applying the sampled natural perturbations to the data samples result in a dataset with bias-conflicting samples.

Another perturbation to consider is the adversarial perturbation $\boldsymbol{\delta}_{f,\epsilon} = \boldsymbol{x}_{adv} - \boldsymbol{x}$ that is generated using f with budget ϵ. Plugging the definition of $\boldsymbol{\delta}_{f,\epsilon}$ into Eq. (1), we have $\boldsymbol{\delta}_{f,\epsilon} = \arg\max_{\|\boldsymbol{\delta}\|\leq\epsilon} \ell(f(\boldsymbol{x} +$

$\boldsymbol{\delta}), y)$. Since the budget ϵ is small, we could approximate the loss function ℓ using the first-order gradient: $\boldsymbol{\delta}_{f,\epsilon} = \arg\max_{\|\boldsymbol{\delta}\| \leq \epsilon} \nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), y)^\top \boldsymbol{\delta} = \epsilon \cdot \frac{\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}),y)}{\|\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}),y)\|}$ [Miyato et al., 2018, Liang et al., 2021]. With the adversarial perturbation, we define the adversarial transferability loss:

$$\ell_{trans}(f_g, f_p, \boldsymbol{x}, y) = \Big(\ell(f_g(\boldsymbol{x} + \boldsymbol{\delta}_{f_g,\epsilon}), y) - \ell(f_g(\boldsymbol{x}), y)\Big) - \Big(\ell(f_p(\boldsymbol{x} + \boldsymbol{\delta}_{f_g,\epsilon}), y) - \ell(f_p(\boldsymbol{x}), y)\Big),$$

which indicates the effectiveness of the adversarial perturbation generated using the global model applied to the personalized models.

C.2 Loss Disparity and Adversarial Transferability

With the definitions of natural and adversarial perturbations, this section shows that both the loss disparity and the adversarial transferability connect to an angle θ. Next, we outline the assumption:

Assumption 1. *The distribution shift does not exacerbate the entanglement of a model f to spurious features $\boldsymbol{x}_s$, which is measured by $\nabla_{\boldsymbol{x}_s} \ell(f(\boldsymbol{x}), y)$:*

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \Delta\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x},y}}[\int_{\alpha=0}^{1} \langle \nabla_{\boldsymbol{x}_s} \ell(f(\boldsymbol{x} + \alpha \cdot \boldsymbol{\Delta}), y)\,, \mathbf{1} \rangle \mathrm{d}\alpha] \leq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[\langle \nabla_{\boldsymbol{x}_s} \ell(f(\boldsymbol{x}), y)\,, \mathbf{1} \rangle].$$

Under Assumption 1, the following Lemmas hold.

Lemma 1. *Under Assumption 1, let Δ be the natural perturbation, θ be the angle between $\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)$ and $\nabla_{\boldsymbol{x}} \ell(f_p(\boldsymbol{x}), y)$, θ_g be the angle between $\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)$ and $\nabla_{\boldsymbol{x}_r} \ell(f_g(\boldsymbol{x}), y) \frown \mathbf{0}$, and γ be $\frac{\|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}),y)\|}{\|\nabla_{\boldsymbol{x}} \ell(f_p(\boldsymbol{x}),y)\|}$, we have:*

$$\begin{aligned} \mathcal{L}(f_p, \mathcal{D}_{bc}) - \mathcal{L}(f_p, \mathcal{D}_b) &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\int_{\alpha=0}^{1} \langle \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x} + \alpha \cdot \boldsymbol{\Delta}), y)\,, \mathbf{1} \rangle \mathrm{d}\alpha] \\ &< \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\frac{\sqrt{d_s}}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot (\sin\theta_g + \sin\theta)] \end{aligned} \tag{4}$$

Lemma 1 connects the loss disparity to θ. The θ_g, differing from θ, is an indicator of the entanglement of the global model to spurious features and is a constant during the personalization step.

Proof. Rewriting $\mathcal{L}(f_p, \mathcal{D}_{bc})$ and introducing $\boldsymbol{\Delta}$:

$$\begin{aligned} \mathcal{L}(f_p, \mathcal{D}_{bc}) &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{bc}}[\ell(f_p(\boldsymbol{x}), y)] \\ &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\ell(f_p(\boldsymbol{x} + \boldsymbol{\Delta}), y)] \\ &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\ell(f_p(\boldsymbol{x}), y) + \ell(f_p(\boldsymbol{x} + \boldsymbol{\Delta}), y) - \ell(f_p(\boldsymbol{x}), y)] \\ &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\ell(f_p(\boldsymbol{x}), y)] \\ &\quad + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\ell(f_p(\boldsymbol{x} + \boldsymbol{\Delta}), y) - \ell(f_p(\boldsymbol{x}), y)] \\ &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[\ell(f_p(\boldsymbol{x}), y)] \\ &\quad + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\ell(f_p(\boldsymbol{x} + \boldsymbol{\Delta}), y) - \ell(f_p(\boldsymbol{x}), y)] \\ &= \mathcal{L}(f_p, \mathcal{D}_b) + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\int_{\alpha=0}^{1} \langle \nabla_{\boldsymbol{x}} \ell(f_p(\boldsymbol{x} + \alpha \cdot \boldsymbol{\Delta}), y)\,, \mathbf{1} \rangle \mathrm{d}\alpha] \\ &= \mathcal{L}(f_p, \mathcal{D}_b) + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\int_{\alpha=0}^{1} \langle \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x} + \alpha \cdot \boldsymbol{\Delta}), y)\,, \mathbf{1} \rangle \mathrm{d}\alpha] \end{aligned}$$

Moving $\mathcal{L}(f_p, \mathcal{D}_b)$ to the left-hand-side (LHS), we have:

$$\mathcal{L}(f_p, \mathcal{D}_{bc}) - \mathcal{L}(f_p, \mathcal{D}_b) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\int_{\alpha=0}^{1} \langle \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x} + \alpha \cdot \boldsymbol{\Delta}), y)\,, \mathbf{1} \rangle \mathrm{d}\alpha]. \tag{5}$$

According to Assumption 1, we further have:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\int_{\alpha=0}^{1} \langle \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x} + \alpha \cdot \boldsymbol{\Delta}), y)\,, \mathbf{1} \rangle] \leq \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_b}[\langle \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}), y)\,, \mathbf{1} \rangle] \tag{6}$$

Next, we connect $\langle \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}+\alpha \cdot \boldsymbol{\Delta}), y)\,, \mathbf{1}\rangle$ to $\|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot \sin\theta$. The first step is connecting $\langle \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}+\alpha \cdot \boldsymbol{\Delta}), y)\,, \mathbf{1}\rangle$ to $\|\nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}), y)\|$ using Cauchy-Schwarz inequality:

$$\begin{aligned}
&\langle \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}+\alpha \cdot \boldsymbol{\Delta}), y)\,, \mathbf{1}\rangle \\
&\leq \sqrt{\langle \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}+\alpha \cdot \boldsymbol{\Delta}), y)\,, \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}+\alpha \cdot \boldsymbol{\Delta}), y)\rangle \cdot \langle \mathbf{1}\,, \mathbf{1}\rangle} \\
&= \sqrt{d_s} \cdot \|\nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}), y)\|
\end{aligned} \tag{7}$$

Then, we connect $\|\nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}), y)\|$ to $\|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\|$. Assuming the global model f_g entangles spurious features and the angle between $\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)$ and $\nabla_{\boldsymbol{x}_r} \ell(f_g(\boldsymbol{x}), y) \frown \mathbf{0}$ is θ_g, we have:

$$\|\nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}), y)\| \leq \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot \sin(\theta_g + \theta). \tag{8}$$

Since it is easy to see that $\theta \in [0, \frac{\pi}{2}]$ and the gradient of $\sin\theta$ is monotonically decreasing in $[0, \frac{\pi}{2}]$, we have:

$$\begin{aligned}
\sin(\theta_g + \theta) &= \int_0^{\theta_g+\theta} \nabla \sin\theta \mathrm{d}\theta \\
&= \int_0^{\theta_g+\theta} \cos\theta \mathrm{d}\theta \\
&< \int_0^{\theta_g} \cos\theta \mathrm{d}\theta + \int_0^{\theta} \cos\theta \mathrm{d}\theta \\
&= \sin\theta_g + \sin\theta
\end{aligned} \tag{9}$$

Combining Eq. (8) and Eq. (9), we have:

$$\|\nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x}), y)\| < \|\nabla_{\boldsymbol{x}} \ell(f_p(\boldsymbol{x}), y)\| \cdot (\sin\theta_g + \sin\theta) \tag{10}$$

Recalling the defition of $\gamma := \frac{\|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\|}{\|\nabla_{\boldsymbol{x}} \ell(f_p(\boldsymbol{x}), y)\|}$ and combining Eq. (5), Eq. (6), Eq. (7), Eq. (10) complete the proof.

□

Lemma 2. *Let ϵ be the attack budget, θ be the angle between $\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)$ and $\nabla_{\boldsymbol{x}} \ell(f_p(\boldsymbol{x}), y)$, γ be $\frac{\|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\|}{\|\nabla_{\boldsymbol{x}} \ell(f_p(\boldsymbol{x}), y)\|}$, and the loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be λ-smooth, twice differentiable, we have*

$$\begin{aligned}
\epsilon \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot (1 - \frac{1}{\gamma} \cdot \cos\theta) - \lambda \cdot \epsilon^2 &\leq \ell_{trans}(f_g, f_p, \boldsymbol{x}, y) \\
\leq \epsilon \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot (1 - \frac{1}{\gamma} \cdot \cos\theta) + \lambda \cdot \epsilon^2 &
\end{aligned} \tag{11}$$

Lemma 2 connects the adversarial transferability loss to θ. In the following analysis, we connect the loss disparity to adversarial transferability via θ.

Proof. Under the definition of the adversarial perturbation, it is easy to see that $\boldsymbol{\delta}_{f,\epsilon} = \epsilon \cdot \frac{\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), y)}{\|\nabla_{\boldsymbol{x}} \ell(f(\boldsymbol{x}), y)\|}$ and $\boldsymbol{\delta}_{f,\epsilon}$ increases the loss by:

$$\begin{aligned}
\ell(f_g(\boldsymbol{x} + \boldsymbol{\delta}_{f_g,\epsilon}), y) - \ell(f_g(\boldsymbol{x}), y) &= \boldsymbol{\delta}_{f_g,\epsilon} \nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y) + \frac{1}{2} \boldsymbol{\delta}_{f_g,\epsilon}^{\top} \nabla_{\tilde{\boldsymbol{x}}_g}^2 \ell(f_g(\tilde{\boldsymbol{x}}_g), y) \boldsymbol{\delta}_{f_g,\epsilon} \\
&= \epsilon \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| + \frac{1}{2} \boldsymbol{\delta}_{f_g,\epsilon}^{\top} \nabla_{\tilde{\boldsymbol{x}}_g}^2 \ell(f_g(\tilde{\boldsymbol{x}}_g), y) \boldsymbol{\delta}_{f_g,\epsilon}
\end{aligned}$$

where $\tilde{\boldsymbol{x}}_g$ is a linear interpolation between $\boldsymbol{x}$ and $\boldsymbol{x} + \boldsymbol{\delta}_{f_g,\epsilon}$, by the Lagrange's mean-value theorem. Similarly, for a transferable adversarial example from f_g applies to f_p, $\boldsymbol{\delta}_{f_g,\epsilon}$ could increase the loss of f_p by:

$$\begin{aligned}
\ell(f_p(\boldsymbol{x} + \boldsymbol{\delta}_{f_g,\epsilon}), y) - \ell(f_p(\boldsymbol{x}), y) &= \boldsymbol{\delta}_{f_g,\epsilon} \nabla_{\boldsymbol{x}} \ell(f_p(\boldsymbol{x}), y) + \frac{1}{2} \boldsymbol{\delta}_{f_g,\epsilon}^{\top} \nabla_{\tilde{\boldsymbol{x}}_p}^2 \ell(f_p(\tilde{\boldsymbol{x}}_p), y) \boldsymbol{\delta}_{f_g,\epsilon} \\
&= \epsilon \cdot \|\nabla_{\boldsymbol{x}} \ell(f_p(\boldsymbol{x}), y)\| \cdot \cos\theta + \frac{1}{2} \boldsymbol{\delta}_{f_g,\epsilon}^{\top} \nabla_{\tilde{\boldsymbol{x}}_p}^2 \ell(f_p(\tilde{\boldsymbol{x}}_p), y) \boldsymbol{\delta}_{f_g,\epsilon}
\end{aligned}$$

where $\cos\theta = \frac{\nabla_{\boldsymbol{x}}\ell(f_g(\boldsymbol{x}),y)\cdot\nabla_{\boldsymbol{x}}\ell(f_p(\boldsymbol{x}),y)}{\|\nabla_{\boldsymbol{x}}\ell(f_g(\boldsymbol{x}),y)\|\|\nabla_{\boldsymbol{x}}\ell(f_p(\boldsymbol{x}),y)\|}$. Plugging the approximations above to the adversarial transferability loss, we have:

$$\begin{aligned}\ell_{trans}(f_g, f_p, \boldsymbol{x}, y) &= \Big(\ell(f_g(\boldsymbol{x}+\boldsymbol{\delta}_{f_g,\epsilon}), y) - \ell(f_g(\boldsymbol{x}), y)\Big) - \Big(\ell(f_p(\boldsymbol{x}+\boldsymbol{\delta}_{f_g,\epsilon}), y) - \ell(f_p(\boldsymbol{x}), y)\Big)\\ &= \epsilon\cdot\|\nabla_{\boldsymbol{x}}\ell(f_g(\boldsymbol{x}), y)\| - \epsilon\cdot\|\nabla_{\boldsymbol{x}}\ell(f_p(\boldsymbol{x}), y)\|\cdot\cos\theta\\ &\quad+\frac{1}{2}\cdot\boldsymbol{\delta}_{f_g,\epsilon}^{\top}\nabla^2_{\tilde{\boldsymbol{x}}_g}\ell(f_g(\tilde{\boldsymbol{x}}_g), y)\boldsymbol{\delta}_{f_g,\epsilon} - \frac{1}{2}\cdot\boldsymbol{\delta}_{f_g,\epsilon}^{\top}\nabla^2_{\tilde{\boldsymbol{x}}_p}\ell(f_p(\tilde{\boldsymbol{x}}_p), y)\boldsymbol{\delta}_{f_g,\epsilon}\end{aligned}$$

Under the λ-smooth assumption on the loss function, the spectral norms of the Hessian metrics are bounded. Therefore, we could bound the norm of the deviate between the quadratic terms [Nesterov, 2003, Proof of Theorem 2.1.5] in the adversarial transferability loss:

$$\|\boldsymbol{\delta}_{f_g,\epsilon}^{\top}\nabla^2_{\tilde{\boldsymbol{x}}_g}\ell(f_g(\tilde{\boldsymbol{x}}_g), y)\boldsymbol{\delta}_{f_g,\epsilon} - \boldsymbol{\delta}_{f_g,\epsilon}^{\top}\nabla^2_{\tilde{\boldsymbol{x}}_p}\ell(f_p(\tilde{\boldsymbol{x}}_p), y)\boldsymbol{\delta}_{f_g,\epsilon}\| \leq 2\lambda\cdot\boldsymbol{\delta}_{f_g,\epsilon}^{\top}\boldsymbol{\delta}_{f_g,\epsilon} = 2\lambda\cdot\epsilon^2 \tag{12}$$

Since the quadratic terms in Eq. (12) are scalars, we have:

$$-2\lambda\cdot\epsilon^2 \leq \boldsymbol{\delta}_{f_g,\epsilon}^{\top}\nabla^2_{\tilde{\boldsymbol{x}}_g}\ell(f_g(\tilde{\boldsymbol{x}}_g), y)\boldsymbol{\delta}_{f_g,\epsilon} - \boldsymbol{\delta}_{f_g,\epsilon}^{\top}\nabla^2_{\tilde{\boldsymbol{x}}_p}\ell(f_p(\tilde{\boldsymbol{x}}_p), y)\boldsymbol{\delta}_{f_g,\epsilon} \leq 2\lambda\cdot\epsilon^2 \tag{13}$$

Plugging Eq. (13) and the definition of γ to $\ell_{trans}(f_g, f_p, \boldsymbol{x}, y)$ completes the proof.

□

C.3 A Generalization Upper Bound

We now present an upper bound of the disparity $\mathcal{L}(f_p, \mathcal{D}_{bc}) - \mathcal{L}(f_p, \mathcal{D}_b)$. The main idea is to derive an upper bound of $\|\nabla_{\boldsymbol{x}}\ell(f_g(\boldsymbol{x}), y)\|\cdot\sin\theta$ in Eq. (4) from Eq. (11).

Theorem 3. *Let $\gamma_{\min}$ be the minimum of γ, with Lemmas 1-2, we have:*

$$\begin{aligned}\mathcal{L}(f_p, \mathcal{D}_{bc}) - \mathcal{L}(f_p, \mathcal{D}_b) < \sqrt{d_s}\cdot\Big(&(\frac{\sin\theta_g+\sqrt{2}}{\gamma_{\min}} - 1)\cdot\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[\|\nabla_{\boldsymbol{x}}\ell(f_g(\boldsymbol{x}), y)\|]\\ &+\frac{1}{\epsilon}\cdot\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[\ell_{trans}(f_g, f_p, \boldsymbol{x}, y)] + \lambda\cdot\epsilon\Big)\end{aligned}$$

Theorem 3 suggests (1) debiasing the global model f_g, whose entanglement to spurious features is measured by θ_g, and (2) enforcing the adversarial transferability between f_g and f_p help reducing the loss disparity of personalized models.

Proof. According to Lemma 2, we know:

$$\ell_{trans}(f_g, f_p, \boldsymbol{x}, y) \geq \epsilon\cdot\|\nabla_{\boldsymbol{x}}\ell(f_g(\boldsymbol{x}), y)\|\cdot(1 - \frac{1}{\gamma}\cdot\cos\theta) - \lambda\cdot\epsilon^2$$

where $\cos\theta = \frac{\nabla_{\boldsymbol{x}}\ell(f_g(\boldsymbol{x}),y)\nabla_{\boldsymbol{x}}\ell(f_p(\boldsymbol{x}),y)}{\|\nabla_{\boldsymbol{x}}\ell(f_g(\boldsymbol{x}),y)\|\|\nabla_{\boldsymbol{x}}\ell(f_p(\boldsymbol{x}),y)\|}$. Then, we derive an upper bound of $\|\nabla_{\boldsymbol{x}}\ell(f_g(\boldsymbol{x}), y)\|\cdot\sin\theta$ from $\ell_{trans}(f_g, f_p, \boldsymbol{x}, y)$. It is easy to see that $\theta\in[0,\pi]$. Therefore, we have:

$$
\begin{aligned}
&\ell_{trans}(f_g, f_p, \boldsymbol{x}, y) \\
&\geq \epsilon \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot (1 - \frac{1}{\gamma} \cdot \mathrm{cos}\theta) - \lambda \cdot \epsilon^2 \\
&= \frac{\epsilon}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot (-\mathrm{cos}\theta) + \epsilon \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| - \lambda \cdot \epsilon^2 \\
&= \frac{\epsilon}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot (\mathrm{sin}\theta - \mathrm{cos}\theta - \mathrm{sin}\theta) + \epsilon \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| - \lambda \cdot \epsilon^2 \\
&= \frac{\epsilon}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot \mathrm{sin}\theta + \frac{\epsilon}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot (\mathrm{cos}\theta - \mathrm{sin}\theta) \\
&\quad + \epsilon \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| - \lambda \cdot \epsilon^2 \\
&\geq \frac{\epsilon}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot \mathrm{sin}\theta + \frac{\epsilon}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot (-\sqrt{2}) \\
&\quad + \epsilon \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| - \lambda \cdot \epsilon^2 \\
&= \frac{\epsilon}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot \mathrm{sin}\theta + \frac{\epsilon \cdot (\gamma - \sqrt{2})}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| - \lambda \cdot \epsilon^2
\end{aligned}
$$

Moving $\|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot \mathrm{sin}\theta$ to the left hand side (LHS):

$$
\begin{aligned}
&\|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot \mathrm{sin}\theta \\
&\quad \leq \frac{\gamma}{\epsilon} \cdot \ell_{trans}(f_g, f_p, \boldsymbol{x}, y) + (\sqrt{2} - \gamma) \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| + \gamma \cdot \lambda \cdot \epsilon
\end{aligned} \tag{14}
$$

According to Lemma 1, we know:

$$
\begin{aligned}
\mathcal{L}(f_p, \mathcal{D}_{bc}) - \mathcal{L}(f_p, \mathcal{D}_b) &= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b, \boldsymbol{\Delta}\sim\mathcal{D}_{\boldsymbol{\Delta}|\boldsymbol{x}}}[\int_{\alpha=0}^{1} \langle \nabla_{\boldsymbol{x}_s} \ell(f_p(\boldsymbol{x} + \alpha \cdot \boldsymbol{\Delta}), y)\, , \mathbf{1} \rangle \mathrm{d}\alpha] \\
&< \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[\frac{\sqrt{d_s}}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot (\mathrm{sin}\theta_g + \mathrm{sin}\theta)]
\end{aligned} \tag{15}
$$

Combining Eq. (15) and Eq. (14), and taking expectation of $\boldsymbol{x}$, y over $\mathcal{D}_b$:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_b}[\frac{\sqrt{d_s}}{\gamma} \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot (\mathrm{sin}\theta_g + \mathrm{sin}\theta)] \\
&\quad \leq \frac{\sqrt{d_s}}{\gamma} \cdot \Big(\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[\|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\| \cdot \mathrm{sin}\theta_g] + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[\frac{\gamma}{\epsilon} \cdot \ell_{trans}(f_g, f_p, \boldsymbol{x}, y)] \\
&\qquad + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[(\sqrt{2} - \gamma) \cdot \|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\|] + \gamma \cdot \lambda \cdot \epsilon \Big) \\
&\quad \leq \sqrt{d_s} \cdot \Big((\frac{\mathrm{sin}\theta_g + \sqrt{2}}{\gamma_{\min}} - 1) \cdot \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[\|\nabla_{\boldsymbol{x}} \ell(f_g(\boldsymbol{x}), y)\|] \\
&\qquad + \frac{1}{\epsilon} \cdot \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_b}[\ell_{trans}(f_g, f_p, \boldsymbol{x}, y)] + \lambda \cdot \epsilon \Big)
\end{aligned} \tag{16}
$$

Plugging Eq. (16) back to Eq. (15) completes the proof.

□