

A APPENDIX: ADDITIONAL EXPERIMENTAL RESULTS ON COMPAS DATASET

We note that for the specific dataset, the training might not converge when λ becomes too large, because the over-focus on reference domain smoothing might affect the model’s accuracy on the data. As highlighted in **red values**, the performance of WFDS becomes worse when $\lambda = 0.9$ on COMPAS dataset. Surprisingly, we found that worst-fair training might hurt the performance of WFDS on the compas data. We believe that this inconsistent and unstable behavior of WFDS due to data scarcity,

Table 4: Robustness study for WFDS with different global domain smoothing strength on COMPAS.

Dataset	λ	$\mathcal{A}_B \uparrow$	$\mathcal{A}_{sub} \downarrow$	$\Phi_D \downarrow$	$\Phi_E \downarrow$
COMPAS	0.1	0.668 ± 0.004	0.262 ± 0.114	0.106 ± 0.027	0.129 ± 0.056
	0.3	0.668 ± 0.006	0.243 ± 0.117	0.098 ± 0.029	0.111 ± 0.061
	0.5	0.667 ± 0.005	0.248 ± 0.110	0.096 ± 0.025	0.106 ± 0.051
	0.7	0.668 ± 0.004	0.192 ± 0.058	0.082 ± 0.009	0.077 ± 0.019
	0.9	0.601 ± 0.078	0.892 ± 0.905	0.052 ± 0.036	0.052 ± 0.049

Table 5: Ablation study of WFDS on COMPAS.

COMPAS	BA \uparrow	A.G. \downarrow	D.P. \downarrow	E.O. \downarrow
WFDS MMD	0.668 ± 0.004	0.192 ± 0.058	0.082 ± 0.009	0.077 ± 0.019
w/o Worst-Fair Training	0.667 ± 0.003	0.164 ± 0.101	0.072 ± 0.021	0.063 ± 0.037
w/o Ref. Domain Smoothing	0.665 ± 00.04	0.323 ± 0.064	0.121 ± 0.015	0.161 ± 0.032

B APPENDIX: TRAINING ALGORITHM: PSEUDOCODE

Algorithm 1: Worst-Fair Domain Smoothing on Client k

```

1 Inputs local data  $\mathbf{X}_k$ , global model  $f_\theta$ , reference distribution  $\mathcal{Q}$ ;
2 Hyperparameters: Number of iteration  $E$ , batch size  $B$ , scaling factor  $\lambda$  adversarial radius  $\epsilon$ , number of
  steps for PGD  $S$  and step size of PGD  $\eta$ ;
3 Output: fair local model  $f_{\theta_k}$ ;
4 Initialize local model  $f_{\theta_k} = f_\theta$ ;
5 for  $e = 1, 2, \dots, E$  do
6   Load a mini-batch  $\mathbf{B}_k$  from  $\mathbf{X}_k$ ;
7   Load a mini-batch  $\mathbf{B}_Q$  from  $\mathcal{Q}$ ;
  // 1. Generate fairness-adversarial examples
8   for  $\mathbf{x}_i \in \mathbf{B}_k$  do
9     Initialize  $\mathbf{x}'_i = \mathbf{x}_i$ ;
10    for  $s = 1, 2, \dots, S$  do
11       $\mathcal{L}(\mathbf{x}'_i) = \hat{\phi}_*(f_{\theta_k}(\mathbf{x}'_i))$ ;
12       $\mathbf{x}_i = \Pi_{\mathbb{B}(\mathbf{x}'_i, \epsilon)}(\mathbf{x}'_i + \eta \text{sign} \nabla_{\mathbf{x}'_i} \hat{\phi}_*(f_{\theta_k}(\mathbf{x}'_i)))$ , where  $\Pi$  is the projection operator;
13    end
14    Replace  $\mathbf{x}_i$  with  $\mathbf{x}'_i$  in  $\mathbf{B}_k$ ;
15  end
  // 2. Compute the fairness-aware loss
16  Compute fairness loss  $\mathcal{L}_{fair}$  with Equation 11 over generated  $\mathbf{x}'$ ;
  // 3. Compute the global smoothing loss
17  Compute MMD loss  $\mathcal{L}_{MMD}$  with Equation 14 with perturbed  $\mathbf{x}'$  and  $\mathbf{B}_Q$ ;
18  Perform backpropagation and update  $f$ ;
  // 4. Update local model
19  Compute overall loss:  $\mathcal{L}_{all} = \mathcal{L}_{fair} + \lambda \mathcal{L}_{MMD}$ ;
20  Perform backpropagation and update  $f_{\theta_k}$ ;
21 end

```
