Primal-Dual+Langevin
Primal-Dual+SVGD
Control+Langevin
Control+SVGD
Fairness Loss
0.030
0.015
0.000
−0.59
−0.63
−0.67
Training LL
Training LL
−0.6
−0.7
−0.8
−0.9
0
1000
2000
Iterations
Disparate Impact
0.14
0.10
0.06
0.842
0.834
0.826
Test Accuracy
Fairness Loss
0.04
0.02
0.00
0
1000
2000
Iterations
Fairness Loss
0.016
0.008
0.000
−0.560
−0.575
−0.590
Training LL
Training LL
−0.6
−0.7
−0.8
−0.9
0
1000
2000
Iterations
Disparate Impact
0.12
0.09
0.06
0.842
0.836
0.830
Test Accuracy
Fairness Loss
0.04
0.02
0.00
0
1000
2000
Iterations
Fairness Loss
0.016
0.008
0.000
−0.60
−0.62
−0.64
Training LL
Training LL
−0.6
−0.7
−0.8
−0.9
0
1000
2000
Disparate Impact
0.12
0.09
0.06
0.832
0.828
0.824
Test Accuracy
Iterations
Fairness Loss
0.04
0.02
0.00
0
1000
2000
Iterations

(a) Trade-off Curve with Different ϵ

(b) Training Curve

(a) Trade-off Curve with Different ϵ

(b) Training Curve

(a) Trade-off Curve with Different ϵ

(b) Training Curve

(1) $\epsilon_t = \epsilon_0$

(2) (1) $\epsilon_t = \epsilon_0(b+t)^{-\gamma}$

(3) Training with Adagrad Optimizer

Figure 4: Results of different methods when using different step size schemes. Column (a): constant $\epsilon_t = \epsilon_0$; Column (b): decaying step size $\epsilon_t = \epsilon_0(b+t)^{-\gamma}$; Column (c): Adagrad (whose master step size we denote by ϵ_0). We search the best ϵ_0 in the grid $[1e-3, 1e-4, 1e-5, 1e-6]$ that achieves the lowest constraint loss at the end of the training. For the other parameters, we use fixed $\gamma = 0.55$ and $b = 1$ for decaying step size, and the default parameters of Adagrad in PyTorch except the master step size ϵ_0.