A. Understanding cross entropy loss in fair adversarial training

As established in the previous sections, we can view the purpose of the adversary's objective function as calculating a test discrepancy between $\mathcal{Z}_0$ and $\mathcal{Z}_1$ for a particular adversary h. Since the adversary is trying to maximize its objective, then a close-to-optimal adversary will have reward $\mathcal{R}_{\mathcal{A}}(h)$ close to the statistical distance between $\mathcal{Z}_0$ and $\mathcal{Z}_1$. Therefore, an optimal adversary can be thought of as regularizing our representations according to their statistical distance. It is essential for our model that the adversary is incentivized to reach as high a test discrepancy as possible, to fully penalize unfairness in the learned representations and in classifiers which may be learned from them.

However, this interpretation falls apart if we use (17) (equivalent to cross entropy loss) as the objective $\mathcal{R}_{\mathcal{A}}(h)$, since it does *not* calculate the test discrepancy of a given adversary h. Here we discuss the problems raised by dataset imbalance for a cross-entropy objective.

Firstly, whereas the test discrepancy is the sum of conditional expectations (one for each group), the standard cross entropy loss is an expectation over the entire dataset. This means that when the dataset is not balanced (i.e. $P(A = 0) \neq P(A = 1)$), the a cross entropy reward will bias the adversary towards predicting the majority class correctly, at the expense of finding a larger test discrepancy. Consider the following toy example: a single-bit representation Z is jointly distributed with sensitive attribute A according to Table 2. Consider the adversary h that predicts A according to $\hat{A}(Z) = T(h(Z))$ where $T(\cdot)$ is a hard threshold at 0.5. Then if h minimizes cross-entropy, then $h^*(0) = \frac{0.03}{0.95}$ and $h^*(1) = \frac{0.02}{0.05}$ which achieves $\mathcal{L}(h) = -0.051$. Thus every Z is classified as $\hat{A} = 0$ which yields test discrepancy $d_h(\mathcal{Z}_0, \mathcal{Z}_1) = 0$. However, if we directly optimize the test discrepancy as we suggest, i.e., $\mathcal{R}_{DP}(h) = d_h(\mathcal{Z}_0, \mathcal{Z}_1)$, $h^*(Z) = Z$, which yields $\mathcal{R}_{\mathcal{DP}}(h) = \mathbb{E}_{A=0}[1 - h] + \mathbb{E}_{A=1}[h] - 1 = \frac{0.92}{0.95} + \frac{0.02}{0.05} - 1 \approx 0.368$ (or vice versa). This shows that a cross-entropy adversarial objective will not, in the unbalanced case, optimize its reward as well as an ℓ_1 objective will.

	$A=0$	$A=1$
$Z=0$	0.92	0.03
$Z=1$	0.03	0.02

Table 2. $p(Z, A)$

B. Training Details

We used single-hidden-layer neural networks for each of our encoder, classifier and adversary, with 20 hidden units for the Health dataset and 8 hidden units for the Adult dataset. We also used a latent space of dimension 20 for Health and 8 for Adult. We train with ℓ_C and r_A as absolute value, as discussed in Section 5, as a more natural relaxation of the binary case for our theoretical results. Our networks used leaky rectified linear units and were trained with Adam (Kingma & Ba, 2014) with a learning rate of 0.001 and a minibatch size of 64, taking one step per minibatch for both the encoder-classifier and the discriminator. When training CLASSLEARN in Algorithm 1 from a learned representation we use a single hidden layer network with half the width of the representation layer, i.e., g. REPRLEARN (i.e., LAFTR) was trained for a total of 1000 epochs, and CLASSLEARN was trained for at most 1000 epochs with early stopping if the training loss failed to reduce after 20 consecutive epochs.

To get the fairness-accuracy tradeoff curves in Figure 2, we sweep across a range of fairness coefficients $\gamma \in [0.1, 4]$. To evaluate, we use a validation procedure. For each encoder training run, checkpoints were made every 50 epochs; r classifiers are trained on each checkpoint (using r different random seeds), and epoch with lowest median error $+\Delta$ on validation set was chosen. We used $r = 7$. Then r more classifiers are trained on an unseen test set. The median statistics (taken across those r random seeds) are displayed.

C. Transfer Learning Table

Table 3. Results from Figure 3 broken out by task. Δ_{EO} for each shown. The transfer task entails identifying a primary condition code that refers to a medical condition. Most fair on each task is bolded. All model names are abbreviated from Figure 3; "TarUnf" is a baseline, unfair predictor learned directly from the target data without fairness specified.

TRA. TASK	TARUNF	TRAUNF	TRAFAIR	TRAY-AF	LAFTR
MSC2A3	0.362	0.370	0.381	0.378	**0.281**
METAB3	0.510	0.579	**0.436**	0.478	0.439
ARTHSPIN	0.280	0.323	0.373	0.337	**0.188**
NEUMENT	0.419	0.419	0.332	0.450	**0.199**
RESPR4	0.181	0.160	0.223	0.091	**0.051**
MISCHRT	0.217	0.213	0.171	0.206	**0.095**
SKNAUT	0.324	**0.125**	0.205	0.315	0.155
GIBLEED	0.189	0.176	0.141	0.187	**0.110**
INFEC4	0.106	0.042	0.026	**0.012**	0.044
TRAUMA	0.020	0.028	0.032	0.032	**0.019**

Since transfer fairness varied much more than accuracy, we break out the results of Fig. 3 in Table 3, showing the fairness outcome of each of the 10 separate prediction tasks. We note that LAFTR provides the fairest predictions on 7 of the 10 tasks, often by a wide margin, and is never too far behind the fairest model for each task. The unfair model TraUnf achieved the best fairness on one task. We suspect this is due to some of these tasks being relatively easy to solve without relying on the sensitive attribute by proxy. Since the equalized odds metric is better aligned with accuracy than demographic parity (Hardt et al., 2016), high accuracy classifiers can sometimes achieve good Δ_{EO} if they do not rely on the sensitive attribute by proxy. Because the data owner has no knowledge of the downstream task, however, our results suggest that using LAFTR is safer than using the raw inputs; LAFTR is relatively fair even when TraUnf is the most fair, whereas TraUnf is dramatically less fair than LAFTR on several tasks.