1. Experiments details

In this section we provide the technical details of our experiments, including the hyperparameter values, the validation setup and the optimization techniques we use.

1.1. Biased MNIST

Following Bahng et al. [2], we use the Adam optimizer with a learning rate of 0.0001, a weight decay of $10^{-4}$ and a batch size of 256. We train for 80 epochs. We do not use any data augmentation scheme. We use 30% of the training set as validation set, and we colorize it using a $\rho$ value of 0.1. The EnD hyperparameters $\alpha$ and $\beta$ are searched using the Bayesian optimization [7] implementation provided by Weights and Biases [3] on the validation set. For $\rho \in \{0.990, 0.995, 0.997\}$, $\alpha$ and $\beta$ are searched in the interval [0; 1], for $\rho = 0.999$ in [0; 50]. To provide a mean performance along with the standard deviation, we select the top 3 models based on the best validation accuracy obtained, and we report the average accuracy on the final test set.

1.2. CelebA

Following Nam et al. [6], we use the Adam optimizer with a learning rate of 0.001, a batch size of 256, and a weight decay of $10^{-4}$. We train for 50 epochs. Images are resized to $224 \times 224$ and augmented with random horizontal flip. To construct the validation set, we sample $N$ images from each pair $(t, b)$ of the training set, where $N$ is 20% the size of the least populated group $(t, b)$. The EnD hyperparameters $\alpha$ and $\beta$ are searched using the Bayesian optimization [7] implementation provided by Weights and Biases [3] on the validation set, in the interval [0; 50]. To provide a mean performance along with the standard deviation, we select the top 3 models based on the best validation accuracy obtained, and we report the average accuracy on the final test sets.

1.3. IMDB Face

We use the Adam optimizer with a learning rate of 0.001, a batch size of 256 and a weight decay of $10^{-4}$. We train for 50 epochs. As with CelebA, images are resized to $224 \times 224$ and randomly flipped at training time for augmentation. In this case, it is not possible to construct a validation set including samples from both EB1 and EB2, without altering the test set composition. Hence, we perform a 4-fold cross validation for every experiment. For example, when training on EB1, we use one fold of EB2 as validation set and the remaining three folds as EB2 test set. We repeat this process until each EB2 fold is used both as validation and as test set. The same process is repeated when training on EB2, by splitting EB1 in validation and test folds. When training for age prediction, we follow Kim et al. [5], by binning the age values in the intervals 0-19, 20-24, 25-29, 30-34, 34-39, 40-44, 45-49, 50-54, 55-59, 60-64, 65-69, 70-100, proposed by Alvi et al. [1]. For every fold, the EnD hyperparameters $\alpha$ and $\beta$ are searched using the Bayesian optimization [7] implementation provided by Weights and Biases [3] on the validation set, in the interval [0; 50]. To provide a mean performance along with the standard deviation, we select the top model for each fold, based on the best validation accuracy obtained. We report the accuracy obtained on the final test sets, as average accuracy among the different folds.

1.4. CORDA

We use standard SGD as optimization technique, with a learning rate of 0.01, decayed by a factor 0.1 every 15 epochs, a weight decay of $10^{-5}$, and a batch size chosen among $\{4, 8, 16, 32\}$. We train for 50 epochs. Images are resized to 500 $\times$ 500 and a center crop is taken, with size 448 $\times$ 448. The network encoder is pre-trained on the publicly available CheXpert [4] dataset. To build the validation set, we use 20% of CORDA-CDSS and 20% of CORDA-SLG. The EnD hyperparameters $\alpha$ and $\beta$ are searched using the Bayesian optimization [7] implementation provided by Weights and Biases [3] on the validation set, in the inter-
val [0: 50]. To provide a mean performance along with the standard deviation, we select the top 3 models based on the best validation accuracy obtained, and we report the average accuracy on the final test sets.

2. Source code

Source code for the EnD technique, including the Biased MNIST example, can be found in the at https://github.com/EIDOSlab/entangling-disentangling-bias.

References


