Supplementary Material

A. Training details

A.1. Transfer learning (finetuned):

For fine-tuning, our implementation follows the TensorFlow tutorial[7]. By using the Keras package in TensorFlow, we can transfer-learning a pretrained Keras network for new classification tasks. Since in our case, the target domains of the datasets are quite different from the source domain of the pretrained model (ImageNet dataset). We use the convolutional layers as our base model. Additionally, we add a fully connected layer following with a dropout layer, and we set the dropout rate to 0.5 before the last classification softmax layer. We find that finetuning every layer performs better than just finetuning the last fully connected layer.

- In Autism Spectrum Disorder (ASD) task, we fine-tune ResNet 50 and Inception V3 using the same pretrained networks as adversarial reprogramming in Section 4. Here, we use Adam with learning rate of 1e-5 as the optimizer and set the batch size to 32. The training epoch is set to 50.
- In Diabetic Retinopathy (DR) and Melanoma detection tasks, we follow the training setting in ASD task and fine-tune three networks, including ResNet 50, Inception V3 and DenseNet 121.

A.2. Training from scratch:

Following the same architectures as finetuning, we train the networks with random initializer on ASD, DR and Melanoma datasets instead of using any pretrained weights. We use Adam as optimizer and set the learning rate to 1e-5 with the batch size of 32. The training epoch is set to 50.

B. Additional experiments

We conducted additional experiments on reprogramming pre-trained ImageNet models for the facial recognition task on Georgia Tech Face Database[8] (GTFD, 50 classes with 15 face images/class for training and 150 images for testing). The results are shown in Table 6. Similar to other tasks show in Section 4, BAR performs much better than baseline methods and is comparable to AR.

Table 6. Test accuracy on facial recognition task.

Model	From scratch	Finetuning	AR	BAR
ResNet 50	95.90%	97.67%	99.29%	**98.32%**
Inpcept. V3	92.11%	95.33%	97.66%	**97.36%**
DenseNet 121	90.12%	91.20%	97.25%	**95.90%**

C. t-SNE

We further visualize data representation of the ASD and DR datasets using t-Distributed Stochastic Neighbor Embedding (t-SNE), a tool to visualize high-dimensional data. We use open-source Scikit-Learn library to implement t-SNE with general perplexity of 50.

For every data point, we extract the hidden representations from the pre-logit layer of ResNet 50 on three datasets, illustrating the difference among "before adversarial reprogramming (AR)", "after AR" and "transfer learning (fine-tuned)". First, for before AR, we pad training data with zero values to fit the input size of source domain. Second, we extract the feature values of input data from transfer learning model with finetuning. Third, for the feature after adversarial reprogramming, we extract the values of original data with the adversarial program.

C.1. Autism Spectrum Disorder classification

As shown in Figure 6, we visualize the data representations of before/after AR and finetuning using t-distributed stochastic neighbor embedding (t-SNE). Colors represent 2 different class labels (ASD/Non-ASD). We can observe that before AR, the data representations are non-separable, whereas after AR they become highly clustered and well separated, leading to high predictability. In contrast, finetuning has worse representation learning performance relative to AR. Adversarial reprogramming indeed learns better data representations for solving the target-domain task.

C.2. Diabetic Retinopathy detection

The t-SNE plot is shown in Figure 7. Adversarial programming learns better data representation than finetuning.

C.3. Melanoma detection

The t-SNE plot is shown in Figure 8. Consistent with the t-SNE plots of ASD and DR tasks, adversarial programming learns better data representation.

[7] https://www.tensorflow.org/tutorials/images/transfer_learning

[8] https://computervisiononline.com/dataset/1105138700

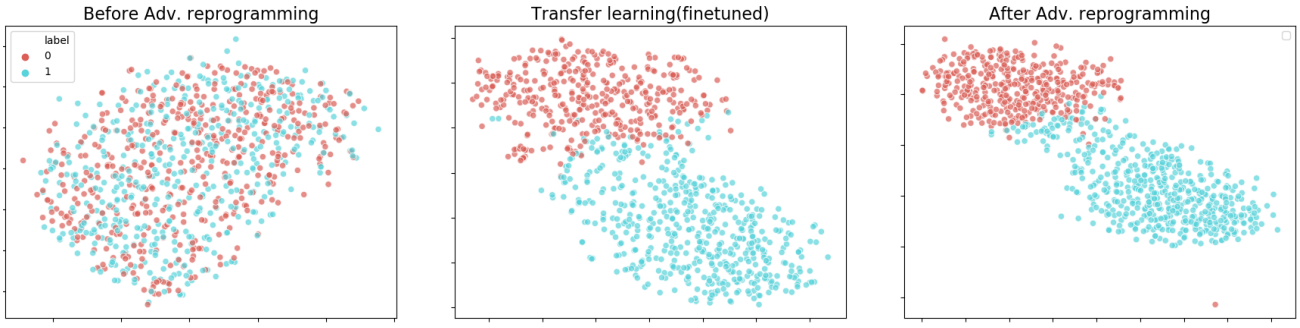

Figure 6. Comparison of t-SNE representations using ResNet 50 and the training data of ASD classification task. Colors represent 2 different class labels.

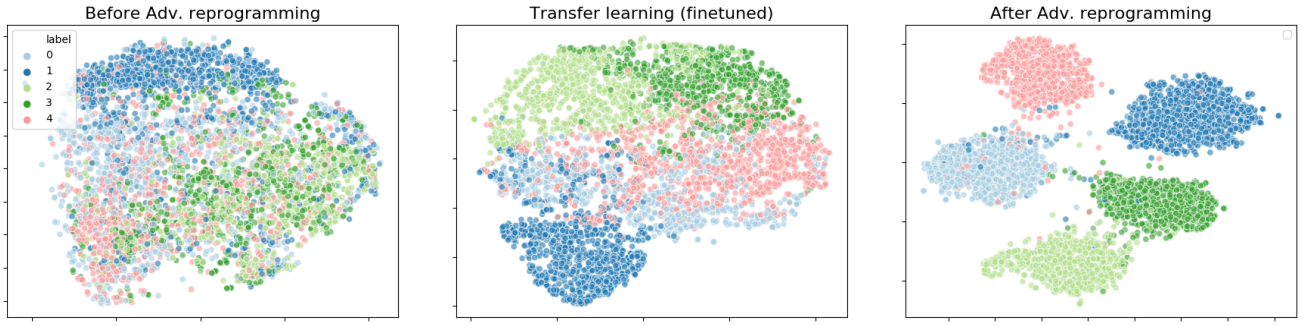

Figure 7. Comparison of t-SNE representations using ResNet 50 and the training data of DR Detection task. Colors represent 5 different class labels.

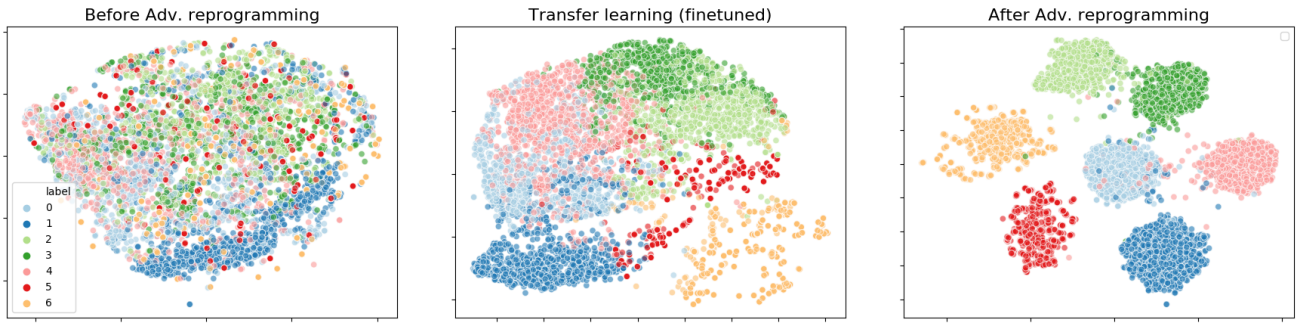

Figure 8. Comparison of t-SNE representations using ResNet 50 and the training data of Melanoma Detection task. Colors represent 7 different class labels.