Supplementary Material for "RxRx1: A Dataset for Evaluating Experimental Batch Correction Methods"

Maciej Sypetkowski[1] Morteza Rezanejad[1] Saber Saberian[1] Oren Kraus[1]
John Urbanik[1] James Taylor[2] Ben Mabey[1] Mason Victors[1] Jason Yosinski[3]
Alborz Rezazadeh Sereshkeh[1] Imran Haque[1] Berton Earnshaw[1]
[1]*Recursion* [2]*Enveda Biosciences* [3]*ML Collective*

1. Additional studies

We include the results of additional experiments in this section.

1.1. Effect of data augmentation and backbone choices

In Table S1 we present perturbation classification accuracy results for different choices of data augmentation methods and convolutional backbones. We note how shallower networks have worse performance, and how AdaBN boosts accuracy over the baseline in all scenarios. We also note how using MixUp instead of CutMix augmentation gives better performance for the baseline but not AdaBN.

Model	Baseline	AdaBN
Default	75.2	87.1
-cutmix	70.8	80.2
-cutmix +mixup	75.6	83.9
backbone=resnet50	71.7	83.9
backbone=resnet101	71.9	84.6
backbone=densenet121	74.3	85.9

Table S1. Perturbation classification accuracy (%) for various choices of data augmentation methods and convolutional backbones of the baseline and AdaBN methods.

1.2. Effect of image normalization methods

In Table S1 we show perturbation classification accuracy for different image normalization methods using our AdaBN method. The preprocessing procedures standardize (*i.e.*, subtract the mean and divide by the standard deviation) each image with the per-channel statistics calculated from different subsets of the dataset. Similar to Table 4, self-standardization (*i.e.*, per-image statistics) offers the best perturbation classification accuracy.

Normalization	Accuracy
All images	78.4
Control images per experiment	83.6
All images per experiment	83.7
Control images per plate	81.7
All images per plate	82.6
Self-standardization	**87.1**

Table S2. Perturbation classification accuracy (%) for different image normalization methods using AdaBN.

1.3. Effect of channel subsets

In Figure S1 we show perturbation classification accuracy of the baseline method trained on all non-empty subsets of channels. Interestingly, we observe that the model that uses all 6 channels does not yield the best performance. All channel subsets containing at least 4 channels without Channel 6 surpass the baseline. Using all but Channel 6 exceeds the baseline by 2 percentage points.

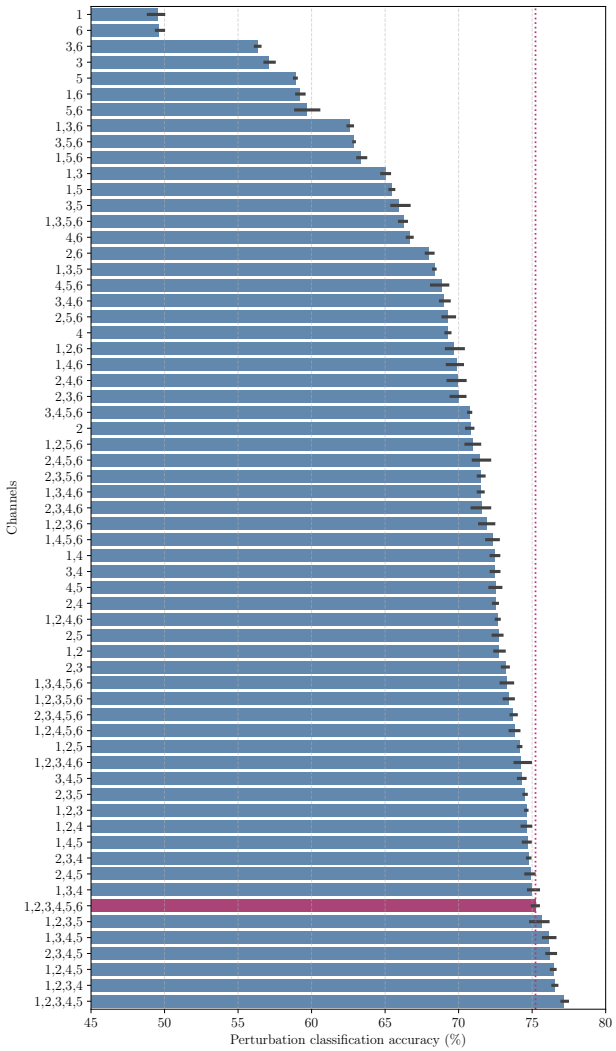

Figure S1. Perturbation classification accuracy of baseline method trained on different subsets of channels.

1.4. Training dynamics

In Figure S2, we plot test perturbation classification accuracy means and standard deviations during model training (over 5 runs). Our results show that the model architecture with AdaBN converges faster than the baseline. Moreover, AdaBN has a much smaller standard deviation than the baseline, *i.e.*, the each run is more consistent with the others for AdaBN.

1.5. Cosine similarity distribution similarity

In Section 5.3, we analyzed the distance between distributions of cosine similarities among image embeddings for both baseline and AdaBN approaches (Figure 7). In Table S3, we provide these distance measures (Wasserstein distance and Kolmogorov-Smirnov statistics) for all batch correction methods.

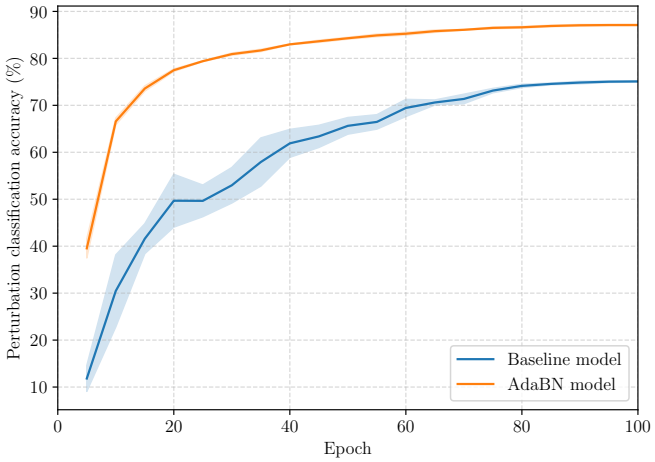

Figure S2. Test perturbation classification accuracy during model training (mean and standard deviation from 5 runs). AdaBN model converges faster, and the runs are more consistent with each other.

Method	Metric	Same perturbation	Different perturbation
Baseline	KS	0.577	0.255
	WD	0.158	0.048
Gradient reversal	KS	0.551	0.198
	WD	0.147	0.036
Adabn	KS	0.210	0.027
	WD	0.051	0.004
AdaBN + gradient reversal	KS	0.202	0.023
	WD	0.052	0.003

Table S3. The Kolmogorov-Smirnov statistic (KS) and Wasserstein distance (WD) between distributions of cosine similarities computed from image embeddings of the same and different perturbations (see Figure 7).