Limitations of Post-Hoc Feature Alignment for Robustness Supplementary Material

1. Additional Experimental Results

1.1. Uncertainty

Recent work has found that models become increasingly less calibrated under distribution shift [6]. Motivated by this problem, we also test whether AdaBN helps with calibration error on the target distribution. We use a simple and popular measure of calibration: the Expected Calibration Error (ECE) [3]. In Table 1 we show the ECE for AdaBN on each dataset. We find that AdaBN substantially reduces calibration error on the corruption benchmarks and Stylized ImageNet, even cutting it in half in most cases.

Table 1: Expected Calibration Error (ECE) of AdaBN and variants on each shifted dataset. AdaBN substantially reduces the ECE on the corruption datasets [4] and Stylized ImageNet.

METHOD	C-10-C	TIN-C	IN-C	INV2	SIN
ORIGINAL	21.5	25.0	12.0	10.6	30.9
ADABN	11.3	15.2	5.2	10.3	12.9
ADABN + AUG	11.7	16.9	6.0	10.9	14.4

1.2. AdaBN on subsets of classes

We now provide additional results showing that applying AdaBN to subsets of classes can hurt accuracy, but that this is mitigated when one does not update the Batch Norm statistics in some of the final layers. In Figure 1 we show the same experiment as in Section 4.1, but this time for TinyImageNet and TinyImageNet-C. The results are qualitatively similar to those for CIFAR-10-C, though the difference between excluding the first layers vs the last layers is less dramatic for TinyImageNet-C.

1.3. The importance of batch information

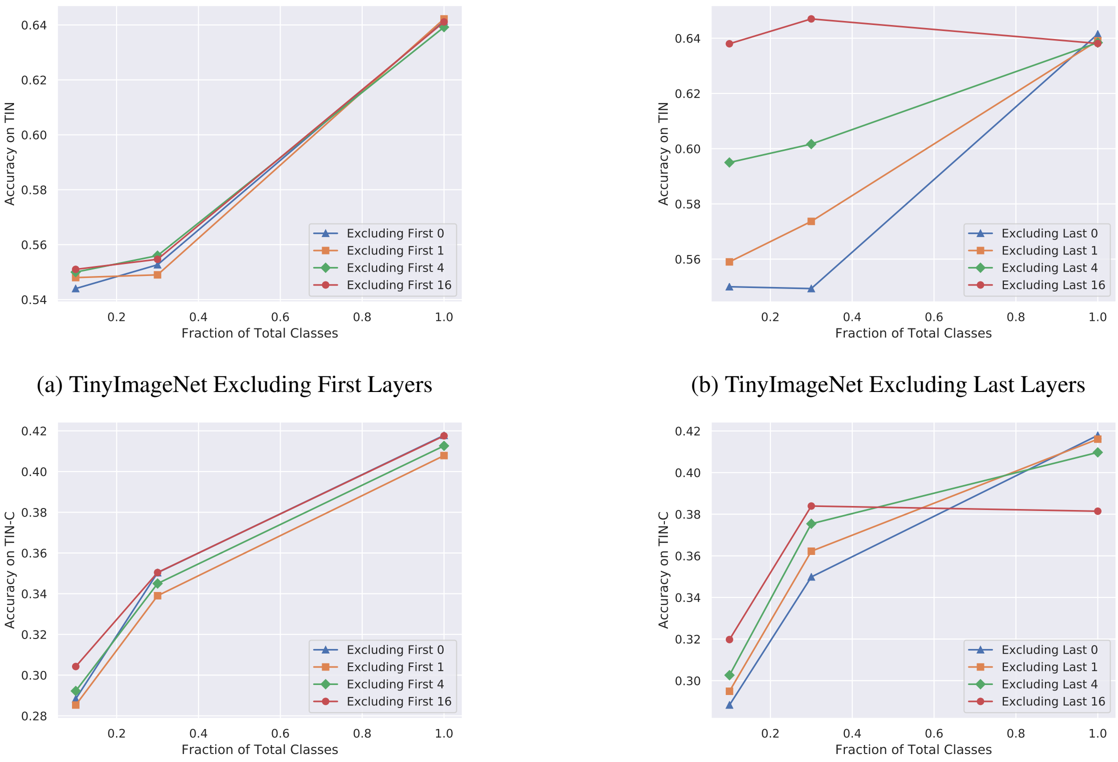
A natural question is whether one can adapt feature alignment methods like AdaBN to a more restricted robustness setting where we do not have access to more than a single example at test time. A simple approach is to use normalization methods other than Batch Norm to align the feature distributions, but which do not use batch information. Two such methods are Group Norm [8] and Instance Norm [7]. Group Norm [8] normalizes over spatial locations and groups of multiple channels within a given layer. Instance Normalization (IN) [7] was introduced for faster stylization. It normalizes over spatial locations over each channel separately, but unlike Group Norm and Batch Norm does not typically include learned affine parameters. We compared models trained using these different normalization schemes on CIFAR-10-C and TinyImageNet-C, along with the corresponding uncorrupted validation sets, and show the results in Table 2. In each case, we use the same architecture and hyperparameters as before, with the only difference being which normalization layer is used. For Group Norm, we test different numbers of groups ranging from 1 to 16, and for Instance Norm we test both with and without learned affine parameters.

We find that the default robustness of the Batch Norm model was much lower on CIFAR-10-C than the default Instance Norm and Group norm models. However, after applying AdaBN to the Batch Norm model, its robustness ended up being higher than the other normalization methods, especially with the augmented version of AdaBN. The results for TinyImageNet are more difficult to interpret because the validation accuracy for Group Norm and especially Instance Norm are worse than for Batch Norm. Still, these results suggest that batch information can be important for improving robustness.

2. Further Discussion of Assumptions

Researchers have attempted to identify assumptions that are sufficient for successful unsupervised domain adaptation. One assumption that has been considered is covariate shift, i.e. $p_S(y|x) = p_T(y|x)$. Ben-David et al. [2] showed that covariate shift is not sufficient for UDA, even when paired with either (i) the assumption that $p_S(x) \approx p_T(x)$ or (ii) the assumption that there is a classifier in the hypothesis class with low error on both domains.

The failures we present can occur even under the covariate shift assumption and even assuming there is no label shift



(a) TinyImageNet Excluding First Layers

(b) TinyImageNet Excluding Last Layers

(c) TinyImageNet-C Excluding First Layers

(d) TinyImageNet-C Excluding Last Layers

Figure 1: The effect of updating the Batch Norm statistics using AdaBN + Aug on different subsets of classes for TinyImageNet and TinyImageNet-C. The results are qualitatively similar to those found in Figure 2 of the main paper.

Table 2: Comparing normalization methods on standard robustness benchmarks. Group Norm and Instance Norm both do worse than Batch Norm under distribution shift, even when the standard test accuracy is comparable.

Method	C-10	C-10-C	TIN	TIN-C
Original model	94.82	72.31	63.80	24.77
AdaBN	92.84	83.63	60.32	40.11
AdaBN + Aug	94.84	86.78	64.05	41.80
IN (no affine)	92.68	81.52	29.54	11.04
IN (affine)	93.51	81.43	45.32	17.04
GN (1 group)	92.53	76.76	56.45	22.14
GN (4 groups)	93.32	78.15	59.34	23.18
GN (16 groups)	93.85	81.68	58.11	22.91

(also known as prior shift or target shift). For shifted spatial locations, this is immediately true because we just made $x_2 = 0$, when x_2 didn't depend on the label in the first place. These two assumptions can also hold for the shifted examples failure; in the simplest case, this is is true when $p(y = -1) = 1$.

The covariate shift assumption is less clear with our failure modes on real data. Nevertheless, it should at least approximately hold in these cases, and can be modified to exactly hold. In particular, while both real shifts (black border and data augmentation) can cut out some relevant features, they rarely change the ground truth label.

3. Theoretical results

3.1. Target error bounds can be uninformative

Denote the target and source classification errors by $\epsilon_T(h)$ and $\epsilon_S(h)$ respectively, and denote the optimal joint error by $\lambda := \min_{h\in\mathcal{H}} \epsilon_T(h) + \epsilon_S(h)$. Ben-David et al. [1] show that for any $h \in \mathcal{H}$,

$$\epsilon_T(h) \leq \epsilon_S(h) + \lambda + |\epsilon_T(h, h^*) - \epsilon_S(h, h^*)| \,, \quad (1)$$

where $\epsilon_S(h, h^*) = Pr_{x\sim D_S}[h(x) \neq h^*(x)]$ and $\epsilon_T(h, h^*) = Pr_{x\sim D_T}[h(x) \neq h^*(x)]$. Ben-David et al. [1] also upper bound $|\epsilon_T(h, h^*) - \epsilon_S(h, h^*)|$ in terms of a distance $d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$ between D_S and D_T,

$$d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) = \sup_{h\in\mathcal{H}} |\epsilon_T(h, h^*) - \epsilon_S(h, h^*)| \,. \quad (2)$$

Many methods aim to minimize $\epsilon_S(h)$ and $d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$. In practice λ is an unknown quantity that depends on the true target labeling function, so most feature alignment methods ignore it. However, this makes it unclear whether this bound provides much of a guarantee even for methods that were directly inspired by it.

We now show that even if one *does* make λ small, such as by using a flexible class $\mathcal{H}$ of neural networks, then the bound proved by Ben-David et al. [1] can be uninformative for a different reason. In particular, when $\lambda = 0$ the bound is equivalent to the triangle inequality. Specifically, when $\lambda = 0$, this means that $\epsilon_T(h, h^*) = \epsilon_T(h)$ and $\epsilon_S(h, h^*) = \epsilon_S(h)$. Hence, the bound in Equation (1) reduces to

$$\epsilon_T(h) \leq \epsilon_S(h) + \left|\epsilon_T(h) - \epsilon_S(h)\right|, \tag{3}$$

which is always true. Upper bounding this in terms of $d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$ is then equivalent to:

$$\epsilon_T(h) \leq \epsilon_S(h) + \sup_{h \in \mathcal{H}} \left|\epsilon_T(h) - \epsilon_S(h)\right|, \tag{4}$$

which is still uninformative.

Other generalization bounds have been proven, such as by Johansson et al. [5], Zhao et al. [9], but these also don't explain why aligning the feature distributions helps in practice. Zhao et al. [9] essentially replace λ with a term that captures the difference between the true source and target labeling functions. Johansson et al. [5] prove a bound based on the support of the source and target distributions that explicitly accounts for the non-invertibility of the feature representation. However, both bounds still include an unobservable quantity that feature alignment methods ignore. Neither paper explains why these unobservable terms should be small in practice for such methods.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. In *Machine learning*, 2010. 2, 3

[2] Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *AISTATS*, 2010. 1

[3] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 1

[4] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 1

[5] Fredrik D Johansson, Rajesh Ranganath, and David Sontag. Support and invertibility in domain-invariant representations. In *AISTATS*, 2019. 3

[6] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019. 1

[7] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 1

[8] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 1

[9] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. In *ICML*, 2019. 3