Test Time Robustification of Deep Models via Adaptation and Augmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of *test time robustification*, i.e., using the test input to improve model robustness. In this work, we aim to study and devise methods that make no assumptions about the model training process and are broadly applicable at test time. We propose a simple approach that can be used in any test setting where the model is probabilistic and adaptable: when presented with a test example, perform different data augmentations on the data point, and then adapt (all of) the model parameters by minimizing the entropy of the model's average, or *marginal*, output distribution across the augmentations. In our experiments, we demonstrate that this approach consistently improves robust ResNet and vision transformer models. We achieve several new state-of-the-art results for test shifts caused by image corruptions (ImageNet-C), renditions of common objects (ImageNet-R), and, among ResNet-50 models, adversarially chosen natural examples (ImageNet-A).

1 Introduction

Robustification of deep models against test distribution shifts is an important and active area of study. We are interested in studying and devising methods for improving model robustness that are "plug and play", i.e., they can be readily used with a wide variety of pretrained models and test settings. In this work, we focus on methods for *test time robustness*, in which the specific test input may be leveraged in order to improve the model's prediction on that point. Though broad applicability is our primary goal, we also want methods that synergize with other robustification techniques. We devise a novel test time robustness method based on adaptation and augmentation: when presented with a test point, we propose to adapt the model by augmenting the test point in different ways and ensuring that the model's predictions are invariant across these augmentations. We further encourage the model to make confident predictions, thus arriving at the proposed method: minimize the *marginal entropy* of the model's predictions across the augmented versions of the test point.

We refer to the proposed method as **m**arginal **e**ntropy **m**inimization with **e**nsembled augmentations (MEME), and this is the primary contribution of our work. MEME makes direct use of pretrained models without any assumptions about their particular training procedure or architecture, while requiring only a single test input for adaptation. In Section 4, we demonstrate empirically that MEME consistently improves the performance of ResNet [7] and vision transformer [5] models on several challenging ImageNet distribution shift benchmarks, achieving several new state-of-the-art results for these models. In particular, MEME consistently outperforms non adaptive marginal distribution predictions (between 1-10% improvement) on corruption and rendition shifts – tested by the ImageNet-C [8] and ImageNet-R [10] datasets, respectively – indicating that adaptation plays a crucial role in improving predictive accuracy. Also, MEME is, to the best of our knowledge, the first adaptation method to improve performance (by 1-4% over standard model evaluation) on ImageNet-A [11], demonstrating that MEME is more broadly applicable on a wide range of shifts.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

2 Related Work

Briefly, most other methods and frameworks that tackle distribution shift problems either assume access to (and modify) the training procedure or adapt using multiple inputs at test time. We provide a more detailed overview of prior work in Appendix A. A few prior works work directly with pretrained models and make minimal test assumptions, similar to MEME. Schneider et al. [29] show that adapting batch normalization (BN) statistics can be effective even with only one test point, and we refer to this approach as "single point" BN adaptation. As we discuss in Section 3 and show empirically in Section 4, MEME synergizes well with single point BN adaptation. Data augmentations are also sometimes used on the test data directly by averaging the model's outputs across augmented copies of the test point [17, 31], i.e., predicting according to the model's marginal output distribution. This technique, which we refer to as test time augmentation (TTA), has been shown to be useful both for improving model accuracy and calibration [1] as well as handling distribution shift [22]. We compare to both single point BN adaptation and TTA in Section 4.

3 Robustness via Adaptation and Augmentation

Data augmentations are typically used to train the model to respect certain invariances – e.g., changes in lighting or viewpoint do not change the underlying class label – but, especially when faced with distribution shift, the model is not guaranteed to obey the same invariances at test time. In this section, we introduce MEME, a method for test time robustness that adapts the model such that it respects these invariances on the test input. We use "test time robustness" specifically to refer to techniques that operate directly on pretrained models and single test inputs – single point BN adaptation and TTA, as described in Section 2, are examples of prior test time robustness methods.

In the test time robustness setting, we are given a trained model $f_\theta : \mathcal{X} \to \mathcal{Y}$ with parameters $\theta \in \Theta$. We do not require any special training procedure and do not make any assumptions about the model, except that θ is adaptable and that f_θ produces a conditional output distribution $p_\theta(y|\mathbf{x})$ that is differentiable with respect to θ. All standard deep neural network models satisfy these assumptions. A single point $\mathbf{x} \in \mathcal{X}$ is presented to f_θ, for which it must predict a label $\hat{y} \in \mathcal{Y}$ immediately. Note that this is precisely identical to the standard test time inference procedure for regular supervised learning models – in effect, we are simply modifying how inference is done, without any additional assumptions on the training process or on test time data availability. This makes test time robustness methods a simple "slot-in" replacement for the ubiquitous and standard test time inference process. We assume sampling access to a set of augmentation functions $\mathcal{A} \triangleq \{a_1, \ldots, a_M\}$ that can be applied to the test point $\mathbf{x}$. Due to the compositionality of many commonly used augmentations, the set size M may be extremely large or even infinite. We use these augmentations and the self-supervised objective detailed below to adapt the model before it predicts on $\mathbf{x}$. When given a set of test data $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the model adapts and predicts on each test point independently, i.e., in the same way as standard non adaptive models. Note that we do not assume access to any ground truth labels.

Given a test point $\mathbf{x}$ and set of augmentation functions $\mathcal{A}$, we sample B augmentations from $\mathcal{A}$ and apply them to $\mathbf{x}$ in order to produce a batch of augmented data $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_B$. The model's average, or *marginal*, output distribution with respect to the augmented points is given by

$$\bar{p}_\theta(y|\mathbf{x}) \triangleq \mathbb{E}_{a \sim \mathcal{U}(\mathcal{A})} \left[p_\theta(y|a(\mathbf{x})) \right] \approx \frac{1}{B} \sum_{i=1}^{B} p_\theta(y|\tilde{\mathbf{x}}_i) \,. \tag{1}$$

What properties do we desire from this marginal distribution? To answer this question, consider the role that data augmentation typically serves during model training. For each training point $(\mathbf{x}^{\text{train}}, y^{\text{train}})$, the model f_θ encounters multiple augmented forms of the input $\tilde{\mathbf{x}}_1^{\text{train}}, \ldots, \tilde{\mathbf{x}}_E^{\text{train}}$, where E usually corresponds to the number of epochs of training, all with the same label y^{train}. In this way, f is trained to obey the invariances between the augmentations and the label – no matter the augmentation, f should predict the same label y^{train} for $\mathbf{x}^{\text{train}}$, and furthermore, it should do so confidently. We seek to devise a similar learning signal during test time, when no ground truth labels are available. That is, we wish to accomplish two goals after adapting the model on a test point:

(1) the model f_θ predictions should be invariant across augmented versions of the test point, and

(2) the model f_θ should be confident in its predictions, even for heavily augmented versions of the test point, due to the additional knowledge that all versions have the same underlying label.

Algorithm 1 Test time robustness via MEME adaptation

Require: trained model f_θ, test point $\mathbf{x}$, # augmentations B, learning rate η, update rule G

1: Sample $a_1, \ldots, a_B \overset{\text{i.i.d.}}{\sim} \mathcal{U}(\mathcal{A})$ and produce augmented points $\tilde{\mathbf{x}}_i = a_i(\mathbf{x})$ for $i \in \{1, \ldots, B\}$

2: Compute Monte Carlo estimate $\tilde{p} = \frac{1}{B}\sum_{i=1}^{B} p_\theta(y|\tilde{\mathbf{x}}_i) \approx \bar{p}_\theta(y|\mathbf{x})$ and $\tilde{\ell} = H(\tilde{p}) \approx \ell(\theta; \mathbf{x})$

3: Adapt model parameters via update rule $\theta' \leftarrow G(\theta, \eta, \tilde{\ell})$

4: Predict $\hat{y} \triangleq \arg\max_y p_{\theta'}(y|\mathbf{x})$

With these two goals in mind, we propose to adapt the model using the entropy of its marginal output distribution over augmentations (Equation 1), i.e.,

$$\ell(\theta; \mathbf{x}) \triangleq H\left(\bar{p}_\theta(\cdot|\mathbf{x})\right) = -\sum_{y \in \mathcal{Y}} \bar{p}_\theta(y|\mathbf{x}) \log \bar{p}_\theta(y|\mathbf{x}) \,. \tag{2}$$

Given that θ is adaptable and $p_\theta(y|\mathbf{x})$ is differentiable with respect to θ, we can directly use gradient based optimization to adapt θ according to this objective. After this step, we can use the adapted model, which we denote $f_{\theta'}$, to predict on the original test input $\mathbf{x}$.

Algorithm 1 presents the overall method MEME for test time adaptation. Though prior test time adaptation methods must carefully choose which parameters to adapt in order to avoid degenerate solutions [33], our adaptation procedure simply adapts all of the model's parameters θ (line 3). The model f_θ adapts using augmented data but makes its final prediction on the original point (line 4).

An additional benefit of MEME is that it synergizes with other approaches for handling distribution shift, thus maximally leveraging the benefits and performance improvements of each technique. As we describe in Appendix B, MEME can be composed with prior methods for training robust models and adapting model statistics. As we show next, these compositions result in improvements over the prior state of the art for several ImageNet distribution shift benchmarks and model architectures.

4 Experiments

We conduct experiments on the ImageNet-C [8], ImageNet-R [10], and ImageNet-A [11] test sets. In all experiments, we compare to single point BN adaptation [29] and the TTA baseline that simply predicts according to the model's marginal output distribution over augmentations $\bar{p}_\theta(y|\mathbf{x})$ (Equation 1) [17, 1]. We also compare to Tent [33], which adapts by minimizing *conditional* entropy, and BN adaptation – both of these methods can be used with pretrained models but require multiple test inputs (or even the entire test set) for adaptation. We provide BN adaptation with 256 test inputs at a time and set additional hyperparameters according to Schneider et al. [29]. For Tent, we use test batch sizes of 64 (note that this is different from the batch size B in Algorithm 1, which denotes how many times a *single* test input is copied) and run online adaptation for the entire epoch through the test set [33]. Full details on our experimental protocol are provided in Appendix C.

We apply MEME on top of the best performing ResNet-50 robust models from prior work, which includes those trained with DeepAugment and AugMix augmentations [10] as well as those trained with moment exchange and CutMix [18]. To evaluate the generality of MEME, we also evaluate the small robust vision transformer (RVT*-small), which provides superior performance on all three ImageNet distribution shift benchmarks compared to the robust ResNet-50 models [21]. Throughout, we use AugMix [9] for our set of augmentation functions $\mathcal{A}$, and we ablate this choice in Appendix D.

We summarize results for ImageNet-C, ImageNet-R, and ImageNet-A in Table 1. We use indentations to indicate composition, e.g., the best results on ImageNet-C for our setting are attained through a combination of starting from a model trained with DeepAugment and AugMix [10] and performing MEME adaptation on top. For both ImageNet-C and ImageNet-R, and for both the ResNet-50 and RVT*-small models, combining MEME with robust training techniques leads to new state of the art performance. We highlight in gray the methods that require multiple test points for adaptation, and we list in bold the best results from these methods which outperform the test time robustness methods. As Table 1 and prior work both show [29, 33], accessing multiple test points can be powerful for benchmarks such as ImageNet-C and ImageNet-R, in which inferred statistics from the test input distribution may aid in prediction. However, these methods do not help, and oftentimes even hurt, for

Table 1: Test results for ImageNet-C, ImageNet-R, and ImageNet-A.

	ImageNet-C mCE ↓	ImageNet-R Error (%)	ImageNet-A Error (%)
Baseline ResNet-50 [7]	76.7	63.9	100.0
+ TTA	77.9 (−1.2)	61.3 (+2.6)	98.4 (+1.6)
+ Single point BN	71.4 (+5.3)	61.1 (+2.8)	99.4 (+0.6)
+ MEME (ours)	69.9 (+6.8)	58.8 (+5.1)	99.1 (+0.9)
+ BN ($n = 256$)	61.6 (+15.1)	59.7 (+4.2)	99.8 (+0.2)
+ Tent [33]	54.4 (+22.3)	57.7 (+6.2)	99.8 (+0.2)
+ DeepAugment+AugMix [10]	53.6	53.2	96.1
+ TTA	55.2 (−1.6)	51.0 (+2.2)	93.5 (+2.6)
+ Single point BN	51.3 (+2.3)	51.2 (+2.0)	95.4 (+0.7)
+ MEME (ours)	**49.8** (**+3.8**)	**49.2** (**+4.0**)	94.8 (+1.3)
+ BN ($n = 256$)	45.4 (+8.2)	48.8 (+4.4)	96.8 (−0.7)
+ Tent [33]	**43.5** (**+10.1**)	**46.9** (**+6.3**)	96.7 (−0.6)
+ MoEx+CutMix [18]	74.8	64.5	91.9
+ TTA	75.7 (−0.9)	62.7 (+1.8)	89.5 (+2.4)
+ Single point BN	71.0 (+3.8)	62.6 (+1.9)	91.1 (+0.8)
+ MEME (ours)	69.1 (+5.7)	59.4 (+3.3)	**89.0** (**+2.9**)
+ BN ($n = 256$)	60.9 (+13.9)	61.6 (+2.9)	93.9 (−2.0)
+ Tent [33]	54.0 (+20.8)	58.7 (+5.8)	94.4 (−2.5)
RVT*-small [21]	49.4	52.3	73.9
+ TTA	53.0 (−3.6)	49.0 (+3.3)	**68.9** (**+5.0**)
+ Single point BN	48.0 (+1.4)	51.1 (+1.2)	74.4 (−0.5)
+ MEME (ours)	**40.6** (**+8.8**)	**43.8** (**+8.5**)	69.8 (+4.1)
+ BN ($n = 256$)	44.3 (+5.1)	51.0 (+1.3)	78.3 (−4.4)
+ Tent [33]	46.8 (+2.6)	50.7 (+1.6)	82.1 (−8.2)

ImageNet-A. Furthermore, we find that these methods are less effective with the RVT*-small model, which may indicate their sensitivity to model architecture choices.

MEME results in substantial improvement for ImageNet-A and is competitive with TTA on this problem. No prior test time adaptation methods have reported improvements on ImageNet-A, and some have reported explicit negative results [29]. As MEME does not rely on batch information and instead extracts a learning signal from single test points, it is, to the best of our knowledge, the first test time adaptation method to report successful results on this testbed. When used on top of a model trained with moment exchange and CutMix [18], MEME achieves state-of-the-art performance among ResNet-50 models. TTA generally offers larger performance gains on ImageNet-A and also results in the highest overall accuracy when combined with the RVT*-small model; however, TTA performs worse than MEME on ImageNet-R and consistently *decreases* accuracy on ImageNet-C.

In Appendix D, we conduct additional experiments on CIFAR-10 [16] distribution shift benchmarks, in which we compare to other baselines and perform ablation studies.

5 Discussion

We presented MEME, a method for test time robustification again distribution shift via adaptation and augmentation. MEME does not require access or changes to the model training procedure and is thus broadly applicable for a wide range of pretrained models. Furthermore, MEME adapts at test time using single test inputs, thus it does not assume access to multiple test points as in several recent methods for test time adaptation [29, 33]. On a range of distribution shift benchmarks for CIFAR-10 and ImageNet classification, and for both ResNet and vision transformer models, MEME consistently improves performance at test time and achieves several new state-of-the-art results for these benchmarks and models. Inference via MEME adaptation is more computationally expensive than standard model inference, and one interesting direction for future work is to develop techniques for selectively determining when to adapt the model in order to achieve more efficient inference.

References

[1] A. Ashukha, A. Lyzhov, D. Molchanov, and D. Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[2] A. Ben-Tal, D. den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 2013.

[3] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[4] G. Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[6] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

[9] D. Hendrycks, N. Mu, E. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations (ICLR)*, 2020.

[10] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[11] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[12] W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning (ICML)*, 2018.

[13] Y. Huang, J. Gornet, S. Dai, Z. Yu, T. Nguyen, D. Tsao, and A. Anandkumar. Neural networks with recurrent generative feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[14] A. Kaku, S. Mohan, A. Parnandi, H. Schambra, and C. Fernandez-Granda. Be like water: Robustness to extraneous variables via adaptive feature normalization. *arXiv preprint arXiv:2002.04019*, 2020.

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[16] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[17] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[18] B. Li, F. Wu, S. Lim, S. Belongie, and K. Weinberger. On feature normalization and data augmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[19] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. Revisiting batch normalization for practical domain adaptation. In *International Conference on Learning Representations Workshop (ICLRW)*, 2017.

[20] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[21] X. Mao, G. Qi, Y. Chen, X. Li, R. Duan, S. Ye, Y. He, and H. Xue. Towards robust vision transformer. *arXiv preprint arXiv:2105.07926*, 2021.

[22] D. Molchanov, A. Lyzhov, Y. Molchanova, A. Ashukha, and D. Vetrov. Greedy policy search: A simple baseline for learnable test-time augmentation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2020.

[23] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning (ICML)*, 2013.

[24] Z. Nado, S. Padhy, D. Sculley, A. D'Amour, B. Lakshminarayanan, and J. Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

[25] A. Orhan. Robustness properties of Facebook's ResNeXt WSL models. *arXiv preprint arXiv:1907.07640*, 2019.

[26] J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[27] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? *arXiv preprint arXiv:1806.00451*, 2018.

[28] S. Sagawa, P. Koh, T. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

[29] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[30] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference (JSPI)*, 2000.

[31] C. Shorten and T. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 2019.

[32] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, 2020.

[33] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

[34] G. Wilson and D. Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2020.

[35] Y. Wu and K. He. Group normalization. In *European Conference on Computer Vision (ECCV)*, 2018.

[36] D. Yin, R. Lopes, J. Shlens, E. Cubuk, and J. Gilmer. A Fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

A Detailed Related Work

The general problem of distribution shift has been studied under a number of frameworks [26], including domain adaptation [30, 4, 34], domain generalization [3, 23, 6], and distributionally robust optimization [2, 12, 28], to name just a few. These frameworks typically leverage additional training or test assumptions in order to make the distribution shift problem more tractable. Largely separate from these frameworks, various empirical methods have also been proposed for dealing with shift, such as increasing the model and training dataset size or using heavy training augmentations [25, 36, 10]. The focus of this work is complementary to these efforts: the proposed MEME method is applicable to a wide range of pretrained models, including those trained via robustness methods, and can achieve further performance gains via test time adaptation.

Prior test time adaptation methods generally either make significant training or test time assumptions. Some methods update the model using batches or even entire datasets of test inputs, such as by computing batch normalization (BN) statistics on the test set [19, 14, 24, 29], or minimizing the (conditional) entropy of model predictions across a batch of test data [33]. The latter approach is closely related to MEME. The differences are that MEME minimizes *marginal* entropy using single test points and data augmentation and adapts all of the model parameters rather than just those associated with normalization layers, thus not requiring multiple test points or specific model architectures. Other test time adaptation methods can be applied to single test points but require specific training procedures or models [32, 13, 29]. Test time training (TTT) [32] requires a specialized model with a rotation prediction head, as well as a different procedure for training this model. As mentioned in Section 2, Schneider et al. [29] show that BN adaptation can be effective at handling some distribution shifts even with only one test point.

A number of works have noted that varying forms of strong data augmentation on the training set can improve the resulting model's robustness [36, 9, 18, 10]. As mentioned in Section 2, data augmentations are also sometimes used on the test data directly by averaging the model's outputs across augmented copies of the test point [17, 31], i.e., predicting according to the model's marginal output distribution. We take this idea one step further by explicitly adapting the model such that its marginal output distribution has low entropy. This extracts an additional learning signal for improving the model, and furthermore, the adapted model can then make its final prediction on the clean test point rather than the augmented copies. We empirically show in Section 4 that these differences lead to improved performance over this non adaptive TTA baseline.

B Composing MEME with Prior Methods

Here, we describe how MEME can be combined with prior methods for training robust models and adapting model statistics.

Pretrained robust models. Since MEME makes no assumptions about, or modifications to, the model training procedure, performing adaptation on top of pretrained robust models, such as those trained with heavy data augmentations, is as simple as using any other pretrained model. Crucially, we find that, in practice, the set of augmentations that we use at test time $\mathcal{A}$ does not have to match the augmentations that were used to train the model. This is important as we require a few properties from the test time augmentations: that they can be easily sampled, ideally can be applied in parallel, and are applied directly to the model input $\mathbf{x}$. These properties do not hold for, e.g., data augmentation techniques based on image translation models, such as DeepAugment [10], or feature mixing, such as moment exchange [18]. However, we can still use models trained with these data augmentation techniques as our starting point for adaptation, thus allowing us to improve upon their state-of-the-art results. As noted above, using pretrained models is not as easily accomplished for adaptation methods which require complicated or specialized training procedures and model architectures, such as TTT [32]. In our experiments, we use AugMix as our set of augmentations [9], as it satisfies the above properties and still yields significant diversity when applied, as depicted in Figure 1.

Adapting BN statistics. Schneider et al. [29] showed that, even when presented with just a single test point, partially adapting the estimated mean and variance of the activations in each batch normalization (BN) layer of the model can still be effective in some cases for handling distribution shift. In this setting, to prevent overfitting to the test point, the channelwise mean and variance

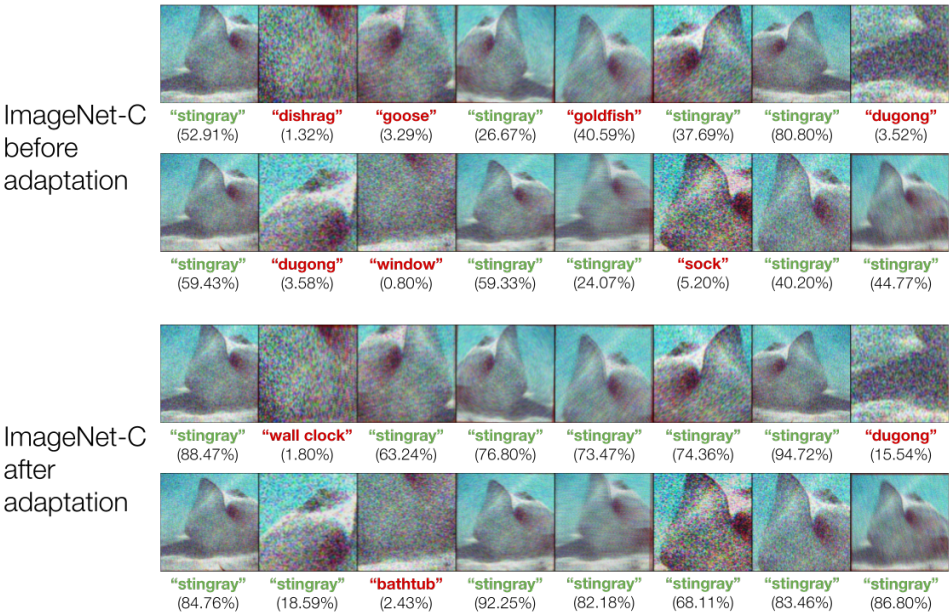

Figure 1: We visualize augmentations of a randomly chosen data point from the "Gaussian Noise level 3" ImageNet-C test set. Even for a robust model trained with heavy data augmentations [10], both its predictive accuracy and confidence drop sharply when encountering test distribution shift. As shown in the bottom row, these drops can be remedied via MEME adaptation.

$[\boldsymbol{\mu}_{\text{test}}, \boldsymbol{\sigma}^2_{\text{test}}]$ estimated from this point are mixed with the the mean and variance $[\boldsymbol{\mu}_{\text{train}}, \boldsymbol{\sigma}^2_{\text{train}}]$ computed during training according to a prior strength N, i.e.,

$$\boldsymbol{\mu} \triangleq \frac{N}{N+1}\boldsymbol{\mu}_{\text{train}} + \frac{1}{N+1}\boldsymbol{\mu}_{\text{test}}\,, \boldsymbol{\sigma}^2 \triangleq \frac{N}{N+1}\boldsymbol{\sigma}^2_{\text{train}} + \frac{1}{N+1}\boldsymbol{\sigma}^2_{\text{test}}\,.$$

This technique is also straightforward to combine with MEME: we simply use the adapted BN statistics whenever computing the model's output distribution $p_{\theta'}$, i.e., both during adaptation (line 2 in Algorithm 1) and when making the final prediction (line 4). Following the suggestion in Schneider et al. [29], we set $N = 16$ for all of our experiments, we specify all of our hyperparameters in the next section.

C Experimental Protocol

We select hyperparameters using the four disjoint validation corruptions provided with ImageNet-C [8]. As the other benchmarks are only test sets and do not provide validation sets, we use the same hyperparameters found using the corruption validation sets and do not perform any additional tuning. For the ResNet-50 models that we evaluate, we use stochastic gradients as the update rule G, and we set $B = 64$ and $\eta = 0.00025$. For the robust vision transformer, we use AdamW [20] as the update rule G, with learning rate $\eta = 0.00001$ and weight decay 0.01, and $B = 64$.

The TTA results are obtained using the same AugMix augmentations as for MEME. The single point BN adaptation results use $N = 16$, as suggested by Schneider et al. [29]. As noted, the BN adaptation results (using multiple test points) are obtained using $N = 256$ as the prior strength and batches of 256 test inputs for adaptation. For Tent, we use the hyperparameters suggested in Wang et al. [33]: we use stochastic gradients with learning rate 0.00025 and momentum 0.9, the adaptation is performed with test batches of 64 inputs, and the method is run online, i.e., prediction and adaptation occur simultaneously and the model is allowed to continuously adapt through the entire test epoch. Since Wang et al. [33] did not experiment with transformer models, we also attempted to run Tent with Adam [15] and AdamW [20] and various hyperparameters for the RVT*-small model; however, we found that this generally resulted in worse performance than using stochastic gradient updates with the aforementioned hyperparameters.

Table 2: Results for the original CIFAR-10 test set, CIFAR-10.1, and CIFAR-10-C. MEME outperforms TTT despite not making any training assumptions. ⋆Results from Sun et al. [32].

	CIFAR-10 Error (%)	CIFAR-10.1 Error (%)	CIFAR-10-C Average Error (%)
ResNet-26 [7]	9.2	18.4	22.5
+ TTA	**7.3** (+1.9)	14.8 (+3.6)	19.9 (+2.6)
+ MEME (ours)	**7.3** (+1.9)	**14.7** (+3.7)	**19.6** (+2.9)
+ Joint training⋆ [32]	8.1	16.7	22.8
+ TTT⋆ [32]	7.9 (+0.2)	15.9 (+0.8)	21.5 (+1.3)

We obtain the baseline ResNet-50 parameters directly from the `torchvision` library. The parameters for the ResNet-50 trained with DeepAugment and AugMix are obtained from `https://drive.google.com/file/d/1QKmc_p6-qDkh51WvsaS9HKFv8bX5jLnP`. The parameters for the ResNet-50 trained with moment exchange and CutMix are obtained from `https://drive.google.com/file/d/1cCvhQKV93pY-jj8f5jITywkB9EabiQDA`. The parameters for the small robust vision transformer (RVT*-small) model are obtained from `https://drive.google.com/file/d/1g4OhuqDVthjS2H5sQV3ppcfcWEzn9ekv`.

D Additional Experiments

We conduct additional experiments on two distribution shift benchmarks for CIFAR-10 [16], specifically, the CIFAR-10-C [8] and CIFAR-10.1 [27] test sets. In these experiments, we compare to test time training (TTT) [32], for which we train ResNet-26 models following their protocol. As mentioned in Appendix A, TTT requires a specialized model and training procedure. As such, we did not compare to TTT for the ImageNet experiments due to the computational demands of training state-of-the-art models and because Sun et al. [32] do not report competitive ImageNet results.

We again select hyperparameters using only the four disjoint validation corruptions provided with CIFAR-10-C [8]. We set the number of augmentations $B = 32$ and the learning rate $\eta = 0.005$, and we again use stochastic gradients as the update rule G. The ResNet-26 model we use for our method closely follows the modifications that Sun et al. [32] propose, in order to provide a fair point of comparison. In particular, Sun et al. [32] elect to use group normalization [35] rather than BN, thus single point BN adaptation is not applicable for this model architecture. As noted before, TTT also requires the joint training of a separate rotation prediction head, thus further changing the model architecture, while MEME directly adapts the standard pretrained model.

D.1 Main results

We summarize results for CIFAR-10, CIFAR-10.1, and CIFAR-10-C in Table 2. We again use indentations to indicate composition, e.g., TTT is performed at test time on top of the joint training procedure. Across all corruption types in CIFAR-10-C, MEME consistently improves test error compared to the baselines, non adaptive TTA, and TTT. MEME also provides a larger performance gain on CIFAR-10.1 compared to TTT. We find that the non adaptive TTA baseline is competitive for these relatively simple test sets, though it is worse than MEME for CIFAR-10-C. Of these three test sets, CIFAR-10-C is the only benchmark that explicitly introduces distribution shift, which suggests that adaptation is useful when the test shifts are more prominent. Both TTA and MEME are also effective at improving performance for the original CIFAR-10 test set where there is no distribution shift, providing further support for the widespread use of augmentations in standard evaluation protocols [17, 1].

D.2 Ablative study

MEME increases model robustness at test time via adaptation and augmentation, and in this section, we ablate both of these axes to understand which design choices are the most important.

Table 3: Ablating the adaptation objective to test pairwise cross entropy and conditional entropy (CE) based adaptation. MEME adaptation generally performs the best, indicating that both encouraging invariance across augmentations and confidence are helpful in adapting the model.

	CIFAR-10 Error (%)	CIFAR-10.1 Error (%)	CIFAR-10-C Average Error (%)
ResNet-26 [7]	9.2	18.4	22.5
+ MEME (ours)	**7.3** (+1.9)	**14.7** (+3.7)	**19.6** (+2.9)
− ℓ (Equation 2) + ℓ_{PCE}	7.4 (+1.8)	14.9 (+3.5)	25.6 (−3.1)
− ℓ (Equation 2) + ℓ_{CE}	7.6 (+1.6)	**14.7** (+3.7)	20.0 (+2.5)

Table 4: Ablating the augmentation functions to test standard augmentations (random resized cropping and horizontal flips). When changing the augmentations used, the post-adaptation performance generally does not change much, though it suffers the most on CIFAR-10-C.

	CIFAR-10 Error (%)	CIFAR-10.1 Error (%)	CIFAR-10-C Average Error (%)
ResNet-26 [7]	9.2	18.4	22.5
+ MEME (ours)	7.3 (+1.9)	14.7 (+3.7)	**19.6** (+2.9)
− AugMix [9] + standard augs	**7.2** (+2.0)	**14.6** (+3.8)	20.2 (+2.3)

Adaptation. From the results above, we conclude that, particularly for more difficult problems such as ImageNet classification, adaptation generally provides additional benefits beyond simply using TTA to predict via the marginal output distribution $\bar{p}_\theta(y|\mathbf{x})$. However, we can disentangle two distinct self-supervised learning signals that may be effective for adaptation: encouraging invariant predictions across different augmentations of the test point, and encouraging confidence via entropy minimization. The marginal entropy objective in Equation 2 encapsulates both of these learning signals, but it cannot easily be decomposed into these pieces. Thus, we instead use two ablative adaptation methods that each only make use of one of these learning signals.

First, we consider optimizing the pairwise cross entropy between each pair of augmented points, i.e.,

$$\ell_{\text{PCE}}(\theta; \mathbf{x}) \triangleq \frac{2}{B \times (B-1)} \sum_{i=1}^{B-1} \sum_{j=i+1}^{B} \sum_{y \in \mathcal{Y}} p_{\text{SG}(\theta)}(y|\tilde{\mathbf{x}}_i) \log p_\theta(y|\tilde{\mathbf{x}}_j) \,,$$

Where $\tilde{\mathbf{x}}_i$ again refers to the i-th sampled augmentation applied to $\mathbf{x}$ and SG is the stop gradient operator. Intuitively, this loss function encourages the model to adapt such that it produces the same predictive distribution for all augmentations of the test point, but it does not encourage the model to produce confident predictions. Conversely, as an objective that encourages confidence but not invariance, we also consider optimizing conditional entropy on the batch of augmented points, i.e.,

$$\ell_{\text{CE}}(\theta; \mathbf{x}) \triangleq \frac{1}{B} \sum_{i=1}^{B} H(p_\theta(\cdot|\tilde{\mathbf{x}}_i)) \,.$$

This ablation is effectively a version of Tent [33] that is suitable for our test setting. That is, instead of assuming a batch of test inputs for adaptation, we instead copy and randomly augment a single test input into a batch. We evaluate these ablations on the CIFAR-10 test sets. We use the same adaptation procedure outlined in Algorithm 1, with ℓ replaced with the above objectives, and we keep the same hyperparameter values $B = 32$ and $\eta = 0.005$.

The results are presented in Table 3. We see that MEME, i.e., marginal entropy minimization, generally performs better than adaptation with either of the alternative objectives. This supports the hypothesis that both invariance across augmentations and confidence are important learning signals for self-supervised adaptation. When faced with more extreme distribution shifts on CIFAR-10-C, we see poor performance from the pairwise cross entropy based adaptation. The conditional entropy based adaptation performs reasonably well on CIFAR-10-C, though worse than MEME.

Augmentation. We also wish to understand the importance of the choice of augmentation functions $\mathcal{A}$. As mentioned, we used AugMix [9] in the previous experiments as it best fit our criteria: AugMix

can be computed in parallel efficiently, requires only the input $\mathbf{x}$, and leads to diverse augmented data points. A simple alternative is to instead use the "standard" set of augmentations commonly used in ImageNet training, i.e., random resized cropping and random horizontal flipping. We evaluate this ablation of using MEME adaptation with standard augmentations also on the CIFAR-10 test sets, again with the same hyperparameter values. From the results in Table 4, we can see that MEME is still effective with simpler augmentation functions. This is true particularly for the cases where there is no test shift, as in the original CIFAR-10 test set, or subtle shifts as in CIFAR-10.1; however, for the more severe and systematic CIFAR-10-C shifts, using heavier AugMix data augmentations leads to greater performance gains over the standard augmentations. Also, this ablation was conducted using the ResNet-26 model, which was trained with standard augmentations, and for robust models such as those in Table 1, AugMix may offer greater advantages at test time since these models were exposed to heavy augmentations during training.