MEMO: Test Time Robustness via Adaptation and Augmentation

Marvin Zhang, Sergey Levine
University of California, Berkeley
{marvin,svlevine}@eecs.berkeley.edu

Chelsea Finn
Stanford University
cbfinn@cs.stanford.edu

Abstract

While deep neural networks can attain good accuracy on in-distribution test points, many applications require robustness even in the face of unexpected perturbations in the input, changes in the domain, or other sources of distribution shift. We study the problem of *test time robustification*, i.e., using the test input to improve model robustness. Recent prior works have proposed methods for test time adaptation, however, they each introduce additional assumptions, such as access to multiple test points, that prevent widespread adoption. 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. Intuitively, this objective encourages the model to make the same prediction across different augmentations, thus enforcing the invariances encoded in these augmentations, while also maintaining confidence in its predictions. In our experiments, we evaluate two baseline ResNet models, two robust ResNet-50 models, and a robust vision transformer model, and we demonstrate that this approach achieves accuracy gains of 1-8% over standard model evaluation and also generally outperforms prior augmentation and adaptation strategies. For the setting in which only one test point is available, we achieve state-of-the-art results on the ImageNet-C, ImageNet-R, and, among ResNet-50 models, ImageNet-A distribution shift benchmarks.

Full paper: `https://arxiv.org/abs/2110.09506`.

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, in order to achieve greater performance than using either set of techniques in isolation. To satisfy both of these desiderata, 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 makes the same predictions across these augmentations, thus respecting the invariances encoded in the data 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.

Workshop on Distribution Shifts, 35th Conference on Neural Information Processing Systems (NeurIPS 2021).

We refer to the proposed method as **m**arginal **e**ntropy **m**inimization with **o**ne test point (MEMO), and this is the primary contribution of our work. MEMO 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 MEMO consistently improves the performance of ResNet [9] and vision transformer [5] models on several challenging ImageNet distribution shift benchmarks, achieving several new state-of-the-art results for these models in the setting in which only one test point is available. In particular, MEMO consistently outperforms non adaptive marginal distribution predictions (between 1-10% improvement) on corruption and rendition shifts – tested by the ImageNet-C [10] and ImageNet-R [12] datasets, respectively – indicating that adaptation plays a crucial role in improving predictive accuracy. Also, MEMO is, to the best of our knowledge, the first adaptation method to improve performance (by 1-4% over standard model evaluation) on ImageNet-A [13], demonstrating the broad applicability of the method.

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 MEMO. Schneider et al. [32] 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, MEMO 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 [19, 34], 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 [25]. 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 MEMO, 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 $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ 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.[1] 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 $\mathbf{x}$. 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 inputs, the model adapts and predicts on each test point independently. 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

[1] Single point BN adaptation also assumes that the model has batch normalization layers, and, as shown empirically in Section 4, this is an assumption that we do not require but can also benefit from.

Algorithm 1 Test time robustness via MEMO

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

marginal, output distribution with respect to the augmented points is given by

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

where the expectation is with respect to uniformly sampled augmentations $a \sim \mathcal{U}(\mathcal{A})$.

What properties do we desire from this marginal distribution? To answer this question, consider the role that data augmentation typically serves during training. For each training point $(\mathbf{x}^{\text{train}}, y^{\text{train}})$, the model f_θ is trained using multiple augmented forms of the input $\tilde{\mathbf{x}}_1^{\text{train}}, \ldots, \tilde{\mathbf{x}}_E^{\text{train}}$. f is trained to obey the invariances between the augmentations and the label – no matter the augmentation on $\mathbf{x}^{\text{train}}$, f should predict the same label y^{train}, and 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, after adapting:

(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.

Optimizing the model for more confident predictions can be justified from the assumption that the true underlying decision boundaries between classes lie in low density regions of the data space [6]. 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.,

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

Optimizing this objective encourages both confidence and invariance to augmentations, since the entropy of $\bar{p}_\theta(\cdot|\mathbf{x})$ is minimized when the model outputs the same (confident) prediction regardless of the augmentation. 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. We use only one gradient step per test point, because empirically we found this to be sufficient for improved performance while being more computationally efficient. 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 MEMO for test time adaptation. Though prior test time adaptation methods must carefully choose which parameters to adapt in order to avoid degenerate solutions [36], our adaptation procedure simply adapts all of the model's parameters θ (line 3). Note that, as discussed above, the model f_θ adapts using augmented data but makes its final prediction on the original point (line 4), which may be easier to predict on.

An additional benefit of MEMO 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, MEMO 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 benchmarks and model architectures in the single test point setting.

4 Experiments

We conduct experiments on the ImageNet-C [10], ImageNet-R [12], and ImageNet-A [13] test sets. In all experiments, we compare to single point BN adaptation [32] and the TTA baseline that simply predicts according to the model's marginal output distribution over augmentations

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

	ImageNet-C mCE ↓	ImageNet-R Error (%)	ImageNet-A Error (%)
Baseline ResNet-50 [9]	76.7	63.9	100.0
+ TTA	77.91.2	61.3 (+2.6)	98.4 (+1.6)
+ Single point BN	71.4 (+5.3)	61.1 (+2.8)	99.4 (+0.6)
+ MEMO (ours)	69.9 (+6.8)	58.8 (+5.1)	99.1 (+0.9)
+ BN ($N = 256, n = 256$)	61.6 (+15.1)	59.7 (+4.2)	99.8 (+0.2)
+ Tent (online) [36]	54.4 (+22.3)	57.7 (+6.2)	99.8 (+0.2)
+ Tent (episodic)	64.7 (+12.0)	61.0 (+2.9)	99.7 (+0.3)
+ DeepAugment+AugMix [12]	53.6	53.2	96.1
+ TTA	55.21.6	51.0 (+2.2)	93.5 (+2.6)
+ Single point BN	51.3 (+2.3)	51.2 (+2.0)	95.4 (+0.7)
+ MEMO (ours)	**49.8 (+3.8)**	**49.2 (+4.0)**	94.8 (+1.3)
+ BN ($N = 256, n = 256$)	45.4 (+8.2)	48.8 (+4.4)	96.80.7
+ Tent (online)	**43.5 (+10.1)**	**46.9 (+6.3)**	96.70.6
+ Tent (episodic)	47.1 (+6.5)	50.1 (+3.1)	96.60.5
+ MoEx+CutMix [20]	74.8	64.5	91.9
+ TTA	75.70.9	62.7 (+1.8)	89.5 (+2.4)
+ Single point BN	71.0 (+3.8)	62.6 (+1.9)	91.1 (+0.8)
+ MEMO (ours)	69.1 (+5.7)	59.4 (+3.3)	**89.0 (+2.9)**
+ BN ($N = 256, n = 256$)	60.9 (+13.9)	61.6 (+2.9)	93.92.0
+ Tent (online)	54.0 (+20.8)	58.7 (+5.8)	94.42.5
+ Tent (episodic)	66.2 (+8.6)	63.9 (+0.6)	94.72.8
RVT*-small [24]	49.4	52.3	73.9
+ TTA	53.03.6	49.0 (+3.3)	**68.9 (+5.0)**
+ Single point BN	48.0 (+1.4)	51.1 (+1.2)	74.40.5
+ MEMO (ours)	**40.6 (+8.8)**	**43.8 (+8.5)**	69.8 (+4.1)
+ BN ($N = 256, n = 256$)	44.3 (+5.1)	51.0 (+1.3)	78.34.4
+ Tent (online)	46.8 (+2.6)	50.7 (+1.6)	82.18.2
+ Tent (adapt all)	44.7 (+4.7)	74.121.8	81.17.2

$\bar{p}_\theta(y|\mathbf{x})$ (Equation 1) [19, 1]. We also compare to Tent [36], 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. [32]. For Tent, we use test batch sizes of 64 and, for ResNet-50 models, test both "online" adaptation – where the model adapts continually through the entire evaluation – and "episodic" adaptation – where the model is reset after each test batch [36]. Note that the evaluation protocols are different for these methods: whereas MEMO is tasked with predicting on each test point immediately after adaptation, BN adaptation predicts on a batch of 256 test points after computing BN statistics on the batch, and Tent predicts on a batch of 64 inputs after adaptation but also, in the online setting, continually adapts throughout evaluation.. Full details on our experimental protocol are provided in Appendix C.

We apply MEMO on top of multiple pretrained models with different architectures. We use the best performing ResNet-50 robust models from prior work, which includes those trained with DeepAugment and AugMix augmentations [12] as well as those trained with moment exchange and CutMix [20]. To evaluate the generality of MEMO, 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 [24]. Throughout, we use AugMix [11] 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 [12] and performing MEMO adaptation on top. For both ImageNet-C and ImageNet-R, and for both the ResNet-50 and

RVT*-small models, combining MEMO with robust training techniques leads to new state-of-the-art performance among methods that observe only one test point at a time. 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 [32, 36], 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. In the case of Tent, which adapts online, the model has adapted using the entire test set by the end of evaluation. However, these methods do not help, and oftentimes even hurt, for 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. Therefore, for this model, we also test a modification of Tent which adapts all parameters, and we find that this version of Tent works better for ImageNet-C but is significantly worse for ImageNet-R.

MEMO 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 [32]. As discussed, it is reasonable for adaptation methods that rely on multiple test points to achieve greater success on other benchmarks such as ImageNet-C, in which a batch of inputs provides significant information about the specific corruption that must be dealt with. In contrast, ImageNet-A does not have such obvious characteristics associated with the input distribution, as it is simply a collection of images that are difficult to classify. As MEMO 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 [20], MEMO achieves state-of-the-art performance among ResNet-50 models and single test point methods. 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 MEMO on ImageNet-R and consistently *decreases* accuracy on ImageNet-C. We view the consistency with which MEMO outperforms the best prior methods, which change across different test sets, to be a major advantage.

In Appendix D, we conduct additional experiments on CIFAR-10 [18] distribution shift benchmarks, in which we compare to other baselines and perform ablation studies. We also present additional ImageNet experiments that test the importance of augmentations, analyze the efficiency to accuracy tradeoff by varying the number of augmentations B, and evaluate ResNext-101 models [39, 23].

5 Discussion

We presented MEMO, a method for test time robustification again distribution shift via adaptation and augmentation. MEMO does not require access or changes to the model training procedure and is thus broadly applicable for a wide range of pretrained models. Furthermore, MEMO 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 [32, 36, 41]. On a range of distribution shift benchmarks, for both ResNet and vision transformer models, MEMO consistently improves performance at test time and achieves several new state-of-the-art results for these models in the single test point setting.

Inference via MEMO is more computationally expensive than standard model inference, primarily because adaptation is performed per test point and thus inference cannot be batched. When deployed in the real world, it is natural to expect that test points will arrive one at a time and batched inference will not be possible. However, MEMO is also more computationally expensive due to its augmentation and adaptation procedure. 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. For example, with well calibrated models [8], we may run simple "feedforward" inference when the prediction confidence is over a certain threshold, thus achieving better efficiency. Additionally, it would be interesting to explore MEMO in the test setting where the model is allowed to continually adapt as more test data is observed. In our preliminary experiments in this setting, MEMO tended to lead to degenerate solutions, e.g., the model predicting a constant label with maximal confidence, and this may potentially be rectified by carefully choosing which parameters to adapt [36] or regularizing the model such that it does not change too drastically from the pretrained model.

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] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

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

[8] C. Guo, G. Pleiss, Y. Sun, and K. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

[9] 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.

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

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

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

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

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

[20] 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.

[21] 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.

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

[23] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision (ECCV)*, 2018.

[24] 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.

[25] 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.

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

[27] 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.

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

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

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

[31] 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.

[32] 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.

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

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

[35] 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.

[36] 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.

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

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

[39] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[40] 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.

[41] M. Zhang, H. Marklund, A. Gupta, S. Levine, and C. Finn. Adaptive risk minimization: Learning to adapt to domain shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

A Detailed Related Work

The general problem of distribution shift has been studied under a number of frameworks [29], including domain adaptation [33, 4, 37], domain generalization [3, 26, 7], and distributionally robust optimization [2, 14, 31], 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 [28, 40, 12]. The focus of this work is complementary to these efforts: the proposed MEMO 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 [21, 16, 27, 32], or minimizing the (conditional) entropy of model predictions across a batch of test data [36]. The latter approach is closely related to MEMO. The differences are that MEMO 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 [35, 15, 32]. Test time training (TTT) [35] requires a specialized model with a rotation prediction head, as well as a different procedure for training this model. Schneider et al. [32] show that BN adaptation 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, MEMO synergizes well with single point BN adaptation.

A number of works have noted that varying forms of strong data augmentation on the training set can improve the resulting model's robustness [40, 11, 20, 12]. Data augmentations are also sometimes used on the test data directly by averaging the model's outputs across augmented copies of the test point [19, 34], 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 [25]. 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 MEMO with Prior Methods

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

Pretrained robust models. Since MEMO 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 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 [12], or feature mixing, such as moment exchange [20]. 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 [35] or ARM [41]. In our experiments, we use AugMix as our set of augmentations [11], as it satisfies the above properties and still yields significant diversity when applied, as depicted in Figure 1.

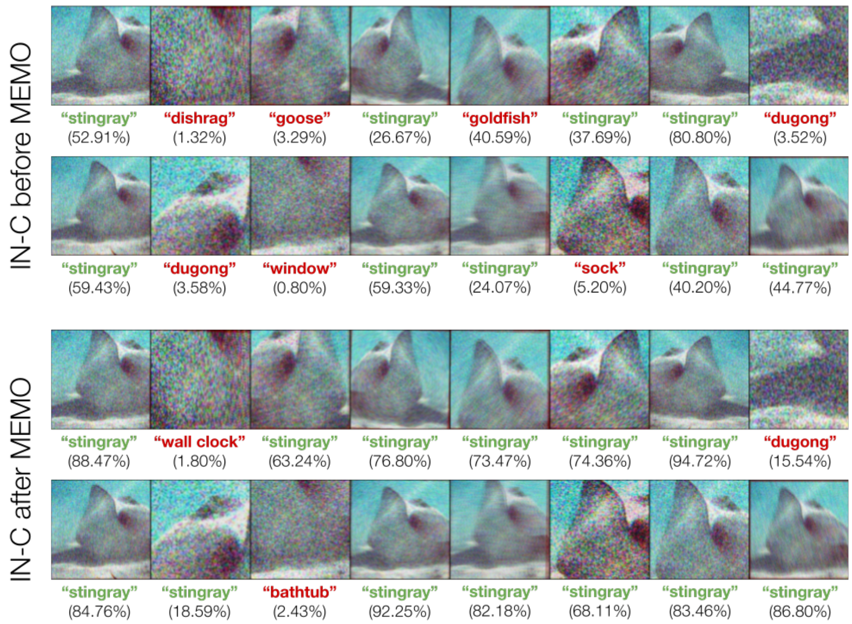

Figure 1: We visualize augmentations of a randomly chosen data point from the "Gaussian Noise level 3" ImageNet-C test set. Even for a robust model trained with heavy data augmentations [12], both its predictive accuracy and confidence drop sharply when encountering test distribution shift. As shown in the bottom two rows, these drops can be remedied via MEMO.

Adapting BN statistics. Schneider et al. [32] 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 $[\boldsymbol{\mu}_{\text{test}}, \boldsymbol{\sigma}^2_{\text{test}}]$ 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 MEMO: we simply use the adapted BN statistics whenever computing the model's output distribution. That is, we adapt the BN statistics alongside all of the model parameters for MEMO. Following the suggestion in Schneider et al. [32], we set $N = 16$ for all of our experiments.

C Experimental Protocol

We select hyperparameters using the four disjoint validation corruptions provided with ImageNet-C [10]. 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 [22] 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 MEMO. The single point BN adaptation results use $N = 16$, as suggested by Schneider et al. [32]. 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. [36]: we use stochastic gradients with learning rate 0.00025 and momentum 0.9, and the adaptation is performed with test batches of 64 inputs. Since Wang et al. [36] did not experiment with transformer models, we also attempted to run Tent with Adam [17] and AdamW [22] 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. MEMO outperforms TTT despite not making any training assumptions. *Results from Sun et al. [35].

	CIFAR-10 Error (%)	CIFAR-10.1 Error (%)	CIFAR-10-C Average Error (%)
ResNet-26 [9]	9.2	18.4	22.5
+ TTA	**7.3** (+1.9)	14.8 (+3.6)	19.9 (+2.6)
+ MEMO (ours)	**7.3** (+1.9)	**14.7** (+3.7)	**19.6** (+2.9)
+ Joint training* [35]	8.1	16.7	22.8
+ TTT* [35]	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/1g40huqDVthjS2H5sQV3ppcfcWEzn9ekv`.

D Additional Experiments

We conduct additional experiments on two distribution shift benchmarks for CIFAR-10 [18], specifically, the CIFAR-10-C [10] and CIFAR-10.1 [30] test sets. In these experiments, we compare to test time training (TTT) [35], 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. [35] do not report competitive ImageNet results.

We again select hyperparameters using only the four disjoint validation corruptions provided with CIFAR-10-C [10]. 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. [35] propose, in order to provide a fair point of comparison. In particular, Sun et al. [35] elect to use group normalization [38] 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 MEMO directly adapts the standard pretrained model.

D.1 CIFAR-10 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, MEMO consistently improves test error compared to the baselines, non adaptive TTA, and TTT. MEMO 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 MEMO 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 MEMO 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 [19, 1].

D.2 Ablative study

MEMO 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. MEMO 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 [9]	9.2	18.4	22.5
+ MEMO (ours)	**7.3** (+1.9)	**14.7** (+3.7)	**19.6** (+2.9)
$-\ \ell$ (Equation 2) + ℓ_{PCE}	7.6 (+1.6)	15.3 (+3.1)	20.0 (+2.5)
$-\ \ell$ (Equation 2) + ℓ_{CE}	7.6 (+1.6)	**14.7** (+3.7)	20.0 (+2.5)
	ImageNet-C mCE ↓	**ImageNet-R Error (%)**	**ImageNet-A Error (%)**
RVT*-small [24]	49.4	52.3	73.9
+ MEMO (ours)	**40.6** (+8.8)	**43.8** (+8.5)	69.8 (+4.1)
$-\ \ell$ (Equation 2) + ℓ_{CE}	41.2 (+8.2)	44.2 (+8.1)	**69.7** (+4.2)

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.,

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

Where $\tilde{\mathbf{x}}_i$ again refers to the i-th sampled augmentation applied to $\mathbf{x}$. 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 the episodic variant of Tent [36] that produces augmented copies of a single test point rather than assuming access to a test batch. We first 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 MEMO, 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 CIFAR-10.1, we see poor performance from the pairwise cross entropy based adaptation method. On the original CIFAR-10 test set and CIFAR-10-C, the ablations perform nearly identically and uniformly worse than MEMO. To further test the ℓ_{CE} ablation, we also evaluate it on the ImageNet test sets for the RVT*-small model. We find that, similarly, minimizing conditional entropy generally improves performance compared to the baseline evaluation. MEMO is more performant for ImageNet-C and ImageNet-R, again indicating the benefits of encouraging invariance to augmentations. Adaptation via ℓ_{CE} performs slightly better for ImageNet-A, though for this problem, TTA is still the best method.

Augmentation. One may first wonder: are augmentations needed in the first place? In the test time robustness setting when only one test point is available, how would simple entropy minimization fare?

Table 4: Evaluating the episodic version of Tent with a batch size of 1, which corresponds to a simple entropy minimization approach for the test time robustness setting. This approach also uses single point BN adaptation, and entropy minimization does not provide much additional gain.

	ImageNet-C mCE ↓	ImageNet-R Error (%)	ImageNet-A Error (%)
Baseline ResNet-50 [9]	76.7	63.9	100.0
+ Single point BN	71.4 (+5.3)	61.1 (+2.8)	99.4 (+0.6)
+ MEMO (ours)	69.9 (+6.8)	58.8 (+5.1)	99.1 (+0.9)
+ Tent (episodic, batch size 1) [36]	70.4 (+6.3)	60.0 (+3.9)	99.3 (+0.7)
+ DeepAugment+AugMix [12]	53.6	53.2	96.1
+ Single point BN	51.3 (+2.3)	51.2 (+2.0)	95.4 (+0.7)
+ MEMO (ours)	**49.8** **(+3.8)**	**49.2** **(+4.0)**	94.8 (+1.3)
+ Tent (episodic, batch size 1)	50.7 (+2.9)	50.7 (+2.5)	95.2 (+0.9)
+ MoEx+CutMix [20]	74.8	64.5	91.9
+ Single point BN	71.0 (+3.8)	62.6 (+1.9)	91.1 (+0.8)
+ MEMO (ours)	69.1 (+5.7)	59.4 (+3.3)	**89.0** **(+2.9)**
+ Tent (episodic, batch size 1)	69.9 (+4.9)	61.7 (+2.8)	90.6 (+1.3)
RVT*-small [24]	49.4	52.3	73.9
+ Single point BN	48.0 (+1.4)	51.1 (+1.2)	74.40.5
+ MEMO (ours)	**40.6** **(+8.8)**	**43.8** **(+8.5)**	**69.8** **(+4.1)**
+ Tent (episodic, batch size 1)	47.9 (+1.5)	50.9 (+1.4)	74.40.5

Table 5: 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 [9]	9.2	18.4	22.5
+ MEMO (ours)	7.3 (+1.9)	14.7 (+3.7)	**19.6** **(+2.9)**
− AugMix [11] + standard augs	**7.2** **(+2.0)**	**14.6** **(+3.8)**	20.2 (+2.3)

We answer this question in Table 4 by evaluating the episodic variant of Tent (i.e., with model resetting after each batch) with a test batch size of 1. This approach is also analogous to a variant of MEMO that does not use augmentations, since for one test point and no augmented copies, conditional and marginal entropy are the same. Similar to MEMO, we also incorporate single point BN adaptation with $N = 16$, in place of the standard BN adaptation that Tent typically employs using batches of test inputs. The results in Table 4 indicate that entropy minimization on a single test point generally provides marginal performance gains beyond just single point BN adaptation. This empirically shows that using augmentations is important for achieving the reported results.

We also wish to understand the importance of the choice of augmentation functions $\mathcal{A}$. As mentioned, we used AugMix [11] in the previous experiments as it best fit our criteria: AugMix requires only the input $\mathbf{x}$, and randomly sampled augmentations lead 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 MEMO with standard augmentations also on the CIFAR-10 test sets, again with the same hyperparameter values. From the results in Table 5, we can see that MEMO 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. Furthermore, this ablation was conducted using the ResNet-26 model, which was trained with standard augmentations – 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.

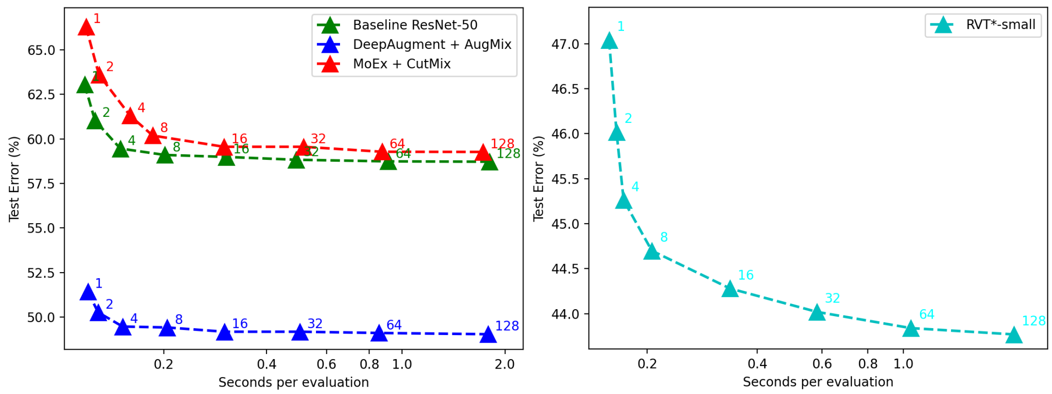

Figure 2: Plotting MEMO efficiency as seconds per evaluation (x axis) and % test error on ImageNet-R (y axis) for the ResNet-50 models (left) and RVT*-small (right) while varying $B = \{1, 2, 4, 8, 16, 32, 64, 128\}$. Note the log scale on the x axis.

Table 6: ImageNet-A results for the ResNext-101 models.

	ImageNet-A Error (%)
Baseline ResNext-101 (32x8d) [39]	90.0
+ TTA	83.2 (+6.8)
+ Single point BN	88.8 (+1.2)
+ MEMO (ours)	84.3 (+5.7)
+ WSL on billions of images [23]	54.9
+ TTA	49.1 (+5.8)
+ MEMO (ours)	**43.2** (**+11.7**)

D.3 Analyzing the tradeoff between efficiency and accuracy

In Figure 2, we analyze the % test error of MEMO adaptation on ImageNet-R as a function of the efficiency of adaptation, measured in seconds per evaluation. We achieve various tradeoffs by varying the number of augmented copies $B = \{1, 2, 4, 8, 16, 32, 64, 128\}$. We note that small values of B such as 4 and 8 can already provide significant performance gains, indicating that a practical tradeoff between efficiency and accuracy is possible. For large B, the wall clock time is dominated by computing the augmentations – in our implementation, we do not compute augmentations in parallel, though in principle this is possible for AugMix and should improve efficiency overall. These experiments used four Intel Xeon Skylake 6130 CPUs and one NVIDIA TITAN RTX GPU.

D.4 Evaluating ResNext101 models on ImageNet-A

ResNext-101 models [39] have been found to achieve higher accuracies on ImageNet-A [13], particularly when trained with massive scale weakly supervised pretraining [23, 12]. In this section, we evaluate whether MEMO can successfully adapt these models and further improve performance on this challenging test set. We use the same hyperparameters as for the robust vision transformer with no additional tuning: AdamW [22] as the update rule G, learning rate $\eta = 0.00001$, weight decay 0.01, and $B = 32$ due to memory limits. We obtain the baseline ResNext-101 (32x8d) parameters, pretrained on ImageNet, directly from the `torchvision` library. We also evaluate a ResNext-101 (32x8d) pretrained with weakly supervised learning (WSL) on billions of Instagram images [23], and we obtained the parameters from `https://download.pytorch.org/models/ig_resnext101_32x8-c38310e5.pth`. For the WSL model, we did not use single point BN adaptation for MEMO as we found this technique to be actually harmful to performance, and this corroborates previous findings [32].

Table 6 summarizes the results. We can see that, similar to Table 1, Both TTA and MEMO significantly improve upon the baseline model evaluation. TTA performs best for the baseline ResNext-101 model. However, MEMO ultimately achieves the best accuracy by a significant margin, as it is more successful at adapting the WSL model, which has a much higher accuracy. This suggests that combining MEMO with other large pretrained models may be an interesting direction for future work.