Compare different Test-time Adaptation methods

7113095004 Liu Chia-Chi

***Abstract*—When we use a pre-trained model, also known as a source model, that has already been trained on labeled source data to make predictions on unseen test data, it is common to observe a degradation in performance. This performance drop is mainly attributed to the domain gap, which refers to the distribution shift between the source domain and the test (target) domain. To mitigate this issue, Test-Time Adaptation (TTA) has emerged as a promising approach. TTA focuses on adapting a model to unlabeled target data during inference, without requiring access to source data or retraining the model from scratch.**

In this paper, we review and contrast several representative TTA methods, including TENT [1] and DeYO [2]. These methods adopt different strategies for online adaptation—TENT performs test-time entropy minimization by updating batch normalization parameters, whereas DISC utilizes discrepancy-based constraints for more stable adaptation. We examine these methods in terms of effectiveness, robustness, and computational cost. Our evaluation aims to provide insights into their practical applicability under varying deployment conditions. Codes can be found on https://github.com/joelleliu17/final_project

***Index Terms*—Test-time Adaptation, TENT, DeYO**

I. Introduction

Traditional machine learning methods typically operate under the assumption that both the training and test data are drawn independently and identically distributed (i.i.d.) from the same underlying distribution [3]. However, this assumption does not hold in real-world scenarios. In practice, the test (or target) distribution frequently differs from the training (or source) distribution, a phenomenon referred to as distribution shift, which can severely degrade the performance of models.

Distribution shifts are pervasive in practical applications. For instance, visual data captured by different types of cameras may vary in resolution or color space [4], road scenes collected from different cities may contain distinct traffic signs and environmental layouts [5], and medical images acquired using different imaging devices across hospitals may differ in contrast or noise characteristics [6]. These domain discrepancies challenge the robustness and generalization capabilities of pre-trained models.

To address such challenges, several research directions have emerged. One such direction is domain generalization (DG) [10][11], which aims to train models that can generalize well to unseen domains by learning domain-invariant representations or enforcing regularization strategies during training. Another closely related area is unsupervised domain adaptation (UDA) [12][13], where models are adapted using unlabeled target domain data alongside labeled source data, typically by aligning feature distributions or leveraging adversarial learning techniques.

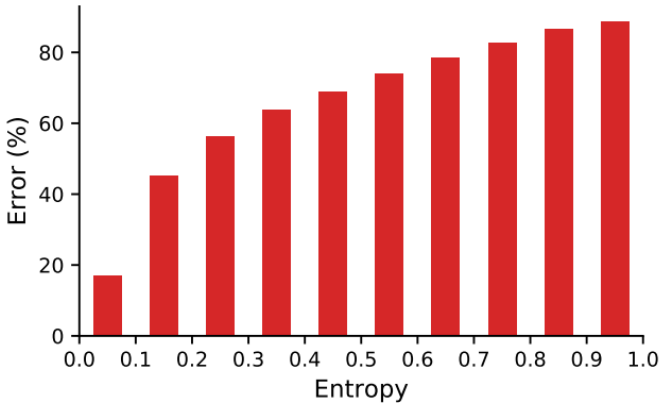


Fig. 1: Predictions with lower entropy have lower error rates on corrupted CIFAR-100-C. Certainty can serve as supervision during testing.

More recently, test-time adaptation (TTA) has gained considerable attention as a practical solution for handling distribution shifts at inference time, where only the unlabeled test data are available and the source data are no longer accessible [1]. In this paper, we focus on two types of test-time adaptation (TTA) methods. By comparing these two categories, we aim to highlight their respective strengths, robustness under domain shifts, and computational costs during inference.

II. Method

A. TENT

TENT (Test-Time Entropy Minimization) [1] is a representative source-free TTA method that adapts a model by minimizing the entropy of its predictions on unlabeled test data. It updates only the batch normalization parameters during inference, allowing the model to become more confident and better aligned with the target domain distribution—without revisiting the source data or altering the original training process. Figure 2 outlines TENT method for fully test-time adaptation.

More specifically, the authors observe that predictions with lower entropy tend to exhibit lower error rates on corrupted datasets such as CIFAR-100-C (Fig.1). Leverage this observation, TENT adapts the model by updating only the affine transformation parameters $\{\gamma_{l,k}, \beta_{l,k}\}$ associated with for each normalization layer l and channel k in the source model. The remaining parameters are fixed. Additionally, the batch normalization statistics $\{\mu_{l,k}, \sigma_{l,k}\}$ from the source data are discarded.

It is important to note that the parameter updates are based solely on the current batch's prediction and therefore only take effect in subsequent batches—unless the forward pass is repeated. TENT supports two adaptation modes:

setting	source data	target data	train loss	test loss
fine-tuning	-	x^t, y^t	$L(x^t, y^t)$	-
domain adaptation	x^s, y^s	x^t	$L(x^s, y^s) + L(x^s, x^t)$	-
test-time training	x^s, y^s	x^t	$L(x^s, y^s) + L(x^s)$	$L(x^t)$
fully test-time adaptation	-	x^t	-	$L(x^t)$

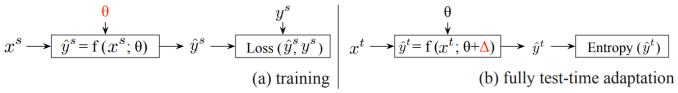


Fig. 2: TENT overview. Tent does not alter training (a), but minimizes the entropy of predictions during testing (b) over a constrained modulation △, given the parameters θ and target data x_t.

- In online adaptation, the model continuously adapts with each incoming batch of test data, without requiring termination criteria.
- In offline adaptation, the model is first adapted using the test set and then inference is performed, optionally over multiple epochs to refine the adaptation.

B. DeYO

Existing TTA methods like Tent[1] often rely on entropy minimization to update the model, under the assumption that low-entropy samples are trustworthy. However, empirical evidence shows that even low-entropy samples can be incorrectly predicted when the model relies on spurious features like the background rather than the object itself(Fig. 3).

To address this limitation, DeYO [2] leverages disentangled latent factors and proposes a novel confidence metric, Pseudo-Label Probability Difference (PLPD), to identify sufficiently reliable test samples for adaptation. As suggested by Geirhos et al. [7], shape is a stable class-prototype-related (CPR) factor because humans—and robust models—rely more on shape than color, and shape is less affected by spurious correlations.

Therefore, the first step is to assess whether each test sample's prediction is based on shape information. This is done by selecting samples that satisfy:

$$S_\theta(x) = \{x \mid \mathrm{Ent}_\theta(x) < \tau_{\mathrm{Ent}}, \quad \mathrm{PLPD}_\theta(x, x') > \tau_{\mathrm{PLPD}}\}, \quad (1)$$

where the PLPD is defined as:

$$\mathrm{PLPD}_\theta(x, x') = \big(p_\theta(x) - p_\theta(x')\big)_{\hat{y}}, \quad (2)$$

with x representing the input, x' its patch-shuffled version, τ_{Ent} and τ_{PLPD} as predefined thresholds, and $\hat{y} = \arg\max p_\theta(x)$ denoting the pseudo-label predicted from x.

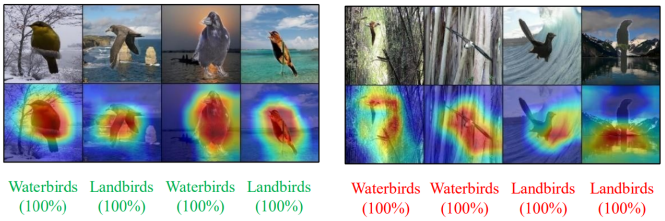
If the model's prediction confidence drops significantly after disrupting the shape structure (i.e., the patch-shuffling), it indicates that the original prediction relied on shape-based features, so the sample can be considered trustworthy for adaptation. For example, consider a test image of a blue car. If models cannot recognize it after patch-shuffling, it indicates that the model relies on the original spatial structure for its prediction, suggesting that this sample is trustworthy for adaptation.

DeYO performs sample selection by exploiting only the samples belonging to $S_\theta(x)$ and calculates the sample-wise weights $alpha_\theta(x)$ to prioritize samples that particularly roots its prediction in the CPR factors. Then, the overall sample-weighted loss is given by:

$$L_{\mathrm{DeYO}}(x; \theta) = \alpha_\theta(x) \cdot \mathbb{I}\{x \in S_\theta(x)\} \cdot \mathrm{Ent}_\theta(x), \quad (3)$$

and $\alpha_\theta(x)$ is:

$$\alpha_\theta(x) = \frac{1}{\exp(\mathrm{Ent}_\theta(x) - \mathrm{Ent}_0)} + \frac{1}{\exp(-\mathrm{PLPD}_\theta(x, x'))} \quad (4)$$



(a) Grad-CAM of correct samples (b) Grad-CAM of wrong samples

Fig. 3: (a) and (b) show Grad-CAM visualizations of samples with correct and incorrect predictions that have very low entropy. When models focus more on the background than the object, prediction errors occur.

III. Experiments

We compare these two methods in terms of performance, effectiveness, and sensitivity to hyperparameters.

Dataset. CIFAR-100-C[8] is a corrupted version of the CIFAR-100 test set, commonly used to evaluate model robustness under distribution shifts. It includes 19 types of common image corruptions (e.g., Gaussian blur, noise, JPEG compression), each applied at 5 severity levels. It simulates the real-world challenges where models encounter data differing from their training distribution.

Test Scenarios and Model. We use ResNet-50 [9] as the baseline model, modifying only the first convolutional kernel to 3×3. The pre-trained model is trained on CIFAR-100 for 100 epochs, achieving a test accuracy of up to 78%. Then we use CIFAR-100-C as test set.

A. Result

Comparison on Different Corruption. As the result in Table.I, under the situation of severity 5, the baseline method achieves only around 30% accuracy in average. TENT shows significant improvement, especially achieving higher accuracy in blur-type corruptions, such as 67.79% accuracy on Zoom blur. DeYO further improves performance and performs noticeably better than TENT on noise-type corruptions—for instance, achieving 10% higher accuracy than TENT on Shot Noise. However, DeYO does not perform well on blur-type corruptions. One possible explanation is that the sample size in CIFAR-100-C is only 32×32, which may be too small for effective recognition. Another assumption is that blur-type corruptions may distort the object's shape, making it harder to identify.

Effectiveness. The results of Table.I and Table.II are base on batch size 64. Under the same setting, TENT takes only

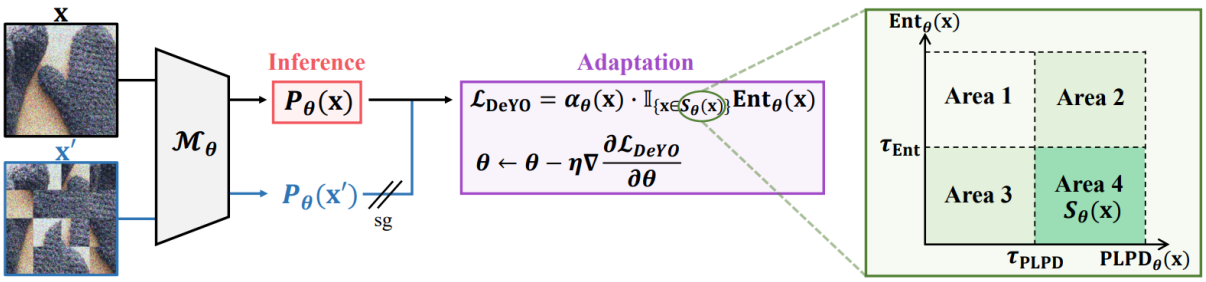


Fig. 4: The overview of DeYO. DeYO comprises sample selection and sample weighting mechanisms. The areas within the green box are distinguished based on entropy and PLPD intervals, with Area 4 corresponding to $S_\theta(x)$.

Method	Gauss	Shot	Impulse	Speckle	Defocus	Glass	Motion	Zoom	G.Blur
No Adapt	10.60	11.76	–	13.25	–	–	–	–	23.32
TENT	39.42	44.39	40.29	49.30	**67.13**	46.99	**64.07**	**67.79**	**65.08**
DeYO	**47.47**	**55.73**	**47.94**	**51.82**	61.81	**50.28**	60.73	64.82	64.50

Method	Bright	Contrast	Elastic	Pixelate	JPEG	Spatter	Saturate	Avg.
No Adapt	66.62	20.40	50.08	24.25	43.11	50.93	58.21	33.89
TENT	**69.83**	64.11	57.82	63.37	54.30	63.52	**68.49**	57.93
DeYO	69.50	**67.71**	**59.61**	**65.02**	**56.90**	**64.49**	67.85	**60.73**

TABLE I: Performance of different methods on various corruptions (severity 5) from CIFAR-100-C, measured in accuracy (%). The best results are in **bold**.

TABLE II: Average running time of each method on 1000 samples.

Method	No Adapt	TENT	DeYO
Time	$<$ 20sec	22sec	32sec

about 5 seconds longer than the baseline, while DeYO takes 32 seconds, which is almost twice as long as the no-adaptation method.

Sensitive. Both TENT and DeYO are sensitive to batch size, as it directly affects the quality of the estimated statistics used for adaptation. For TENT, which updates model parameters based on entropy minimization, a small batch size may lead to unstable gradient estimates and poor entropy approximations, reducing its effectiveness. Similarly, a part of DeYO's adaptation process—such as estimating the PLPD (Equ.2) and computing reliable sample weights—also depends on batch-level statistics. When the batch size is too small, the PLPD estimation becomes noisy and less reliable, making it harder to accurately identify CPR-based (reliable) samples. This can lead to suboptimal sample selection and inaccurate weighting, thereby diminishing DeYO's overall adaptation performance. Thus, like TENT, DeYO requires sufficiently large batch sizes to ensure stable and meaningful adaptation. Additionally, smaller batch sizes can lead to longer adaptation time, as more iterations are needed to process the same amount of data.

IV. Conclusion

DeYO and TENT are two effective test-time adaptation (TTA) methods. These methods adapt the model by updating the batch normalization (BN) layers, which allows for fast adaptation and enables online testing in real-world scenarios. However, this adaptation strategy relies entirely on the presence of BN layers; if a model lacks BN layers, these methods cannot perform adaptation. Moreover, both methods require a sufficiently large number of samples during testing to achieve optimal performance. This means that if samples arrive sequentially, one by one, their adaptation capability is very limited and they can only make minimal adjustments. This limitation poses challenges for applications where only a few samples are available at test time or when real-time adaptation with minimal latency is required.

To address these limitations, future work could explore and compare alternative branches of test-time adaptation (TTA) methods that do not rely solely on batch normalization. For instance, distribution matching, self-training, or pseudo-labeling-based approaches may offer greater flexibility in settings with limited batch size or no BN layers.

References

[1] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *Proc. ICLR*, 2021.

[2] J. Lee, D. Jung, S. Lee, J. Park, J. Shin, U. Hwang, and S. Yoon, "Entropy is not enough for test-time adaptation: From the perspective of disentangled factors," in *Proc. ICLR*, 2024.

[3] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2008.

[4] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. ECCV*, 2010, pp. 213–226.

[5] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. F. Wang, and M. Sun, "No more discrimination: Cross-city adaptation of road scene segmenters," in *Proc. ICCV*, 2017, pp. 1992–2001.

[6] X. Liu and Y. Yuan, "A source-free domain adaptive polyp detection framework with style diversification flow," *IEEE Trans. Med. Imaging*, vol.41, no.7, pp.1897–1908, 2022.

[7] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-Trained CNNs Are Biased Towards Texture; Increasing Shape Bias Improves Accuracy and Robustness," in *Proc. ICLR*, 2019.
[8] D. Hendrycks and T. Dietterich, "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations," in *Proc. ICLR*, 2019.
[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
[10] 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*, vol.24, 2011.
[11] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," in *Proc. ICLR*, 2021.
[12] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, 2015, pp. 1180–1189.
[13] C. Park, J. Lee, J. Yoo, M. Hur, and S. Yoon, "Joint contrastive learning for unsupervised domain adaptation," *arXiv preprint arXiv:2006.10297*, 2020.