On Pitfalls of Test-time Adaptation

Hao Zhao, Yuejiang Liu, Alexandre Alahi
EPFL
{hao.zhao, yuejiang.liu, alexandre.alahi}@epfl.ch

Tao Lin*
Westlake University
lintao@westlake.edu.cn

Abstract

Test-Time Adaptation (TTA) has recently gained significant attention as a new paradigm for tackling distribution shifts. Despite the sheer number of existing methods, the inconsistent experimental conditions and lack of standardization in prior literature make it difficult to measure their actual efficacies and progress. To address this issue, we present a large-scale open-sourced Test-Time Adaptation Benchmark, dubbed TTAB, which includes nine state-of-the-art algorithms, a diverse array of distribution shifts, and two comprehensive evaluation protocols. Through extensive experiments, we identify three common pitfalls in prior efforts: (i) choosing appropriate hyper-parameter, especially for model selection, is exceedingly difficult due to online batch dependency; (ii) the effectiveness of TTA varies greatly depending on the quality of the model being adapted; (iii) even under optimal algorithmic conditions, existing methods still systematically struggle with certain types of distribution shifts. Our findings suggest that future research in the field should be more transparent about their experimental conditions, ensure rigorous evaluations on a broader set of models and shifts, and re-examine the assumptions underlying the potential success of TTA for practical applications.

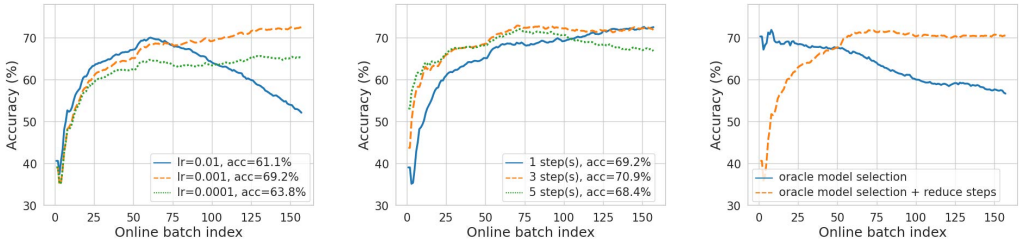
1 Introduction

Tackling the robustness issue under distribution shifts is one of the most pressing challenges in machine learning (Koh et al., 2021). Among existing approaches, Test-Time Adaptation (TTA)—in which neural network models are adapted to new distributions by making use of unlabeled examples at test time—has been an emerging paradigm of growing popularity (Lee et al., 2022; Kundu et al., 2022; Gong et al., 2022a; Chen et al., 2022; Goyal et al., 2022; Sinha et al., 2023). Recent works have proposed a broad array of unsupervised proxy objectives, ranging from entropy minimization (Wang et al., 2021) and self-supervised learning (Sun et al., 2020) to pseudo-labeling (Liang et al., 2020) and feature alignment (Liu et al., 2021), to name a few. Nevertheless, the efficacy of TTA in practical settings as well as the actual progress in the field are often called into question due to inconsistent experimental conditions and a lack of standardization in prior literature (Boudiaf et al., 2022; Su et al., 2022).

The goal of this work is to gain a thorough understanding of the current state of TTA methods while setting the stage for critical problems to be worked on. To this end, we present TTAB, a large-scale open-sourced Test-Time Adaptation Benchmark (see details in appendix E), featuring rigorous evaluations, comprehensive analyses as well as extensive baselines. Our benchmark carefully examines nine state-of-the-art TTA algorithms on a wide range of distribution shifts, under two evaluation protocols. In particular, we place a strong emphasis on some subtle yet essential experimental settings from the practical standpoint that have been largely overlooked in previous works. Our analyses unveil three common pitfalls in prior TTA methods:

Pitfall 1: Hyperparameters have a strong influence on the effectiveness of TTA, and yet they are exceedingly difficult to choose in practice without prior knowledge of the properties and structures of distribution shifts. Our results show that the common practice of

*Corresponding author



(a) Batch dependency exists in the online TTA setting with a single adaptation step.

(b) Multiple-step improves TTA but still has strong dependency among batches.

(c) Oracle model selection may introduce a more serious dependency problem to TTA.

Figure 1: **The batch dependency issue during TTA and non-trivial model selection**, for evaluating SHOT on CIFAR10-C (gaussian noise). Similar trends can be found in all corruption types.

hyperparameter choice for TTA methods does not necessarily improve test accuracy. Instead, it can possibly lead to detrimental effects. Moreover, we find that even when the labels of test examples are available, selecting the TTA hyperparameters for model selection remains challenging, largely due to batch dependency during online adaptation.

Pitfall 2: Even if hyperparameters are optimally selected given oracle information in the test domain, the effectiveness of TTA is not equal on different models. In fact, the degree of improvement strongly depends on the quality of the pre-trained model, not only on its accuracy in the source domain but also on its feature properties. Crucially, we find that good practice in data augmentations (Hendrycks et al., 2019; 2022) for out-of-distribution generalization leads to reverse effects for TTA.

Pitfall 3: Even under ideal conditions where optimal hyperparameters are used in conjunction with suitable pre-trained models, existing methods still perform poorly on certain classes of distribution shifts, such as correlation shifts (Sagawa et al., 2019) and label shifts (Sun et al., 2022)), that are infrequently considered in the realm of TTA but widely used in domain adaptation and domain generalization. This limitation, combined with the previously mentioned issues, raises significant concerns regarding the feasibility and potential of TTA in addressing distribution shifts that occur in the natural world beyond our control.

Aside from these empirical results obtained, our TTAB benchmark is designed as an expandable package that standardizes experimental settings while easing the incorporation of novel algorithmic implementations with minimal difficulty. We hope the accompanied library will not only facilitate fair and transparent empirical evaluations of new algorithms across a broader range of base models and distribution shifts, but also stimulate further insights into the assumptions that underlie the potential utility of TTA in practical applications. Our code and leaderboards will be publicly available.

2 The Limits of Evaluation for TTA Methods

Despite the existence of various TTA methods stated in appendix A, their effectiveness and feasibility in practice are unknown due to inconsistent adaptation setups. This section starts by thoroughly reviewing representative TTA methods on standard CIFAR10-C (Hendrycks & Dietterich, 2019) in appendix C.1, which points out TTA methods are highly sensitive to the choices of hyperparameters and inconsistent adaptation setups across methods further hinder fair evaluation. The identified issue of improper evaluation in prior work motivates us to compare the optimal test-time performance for each method, where we surprisingly find it is non-trivial: 1) the oracle model selection does not exist in TTA due to the batch dependency (c.f. §2.1), and 2) even if hyperparameters are optimally selected, the effectiveness of TTA methods varies based on pre-trained model qualities (see §2.2).

2.1 Enabling Fair Evaluation During TTA

2.1.1 The Batch Dependency Issues During TTA

Most existing TTA methods, as identified in Table 4, tend to leverage distribution knowledge (i.e. adaptation history) learned from previous test batches to improve the test-time performance on new samples, in which the choice of online is dominant over that of episodic. This design inherently increases the risk of batch dependency during TTA.

The existence of batch dependency. Figure 1(a) examines the common online setting with a single adaptation step and a range of learning rates: *the batch dependency issue is only diminished for relatively small values of the learning rate*, otherwise, a considerable descent in performance can be observed as TTA progresses. Moving to the phase of multiple adaptation steps with a relatively small learning rate in Figure 1(b), we observe that adaptation performance increases from 69.2% to 70.9%. However, if we continue to increase the number of adaptation steps, the adaptation performance quickly drops to 68.4% *due to over-adaptation on previous test batches.*

In conclusion, dependency on adaptation history is a common issue in dominant TTA methods, making adaptation on subsequent test samples difficult, particularly when using inappropriate hyperparameters. This phenomenon is particularly pronounced in the online TTA setting with varying hyperparameter combinations. In §2.1.2, we will further elaborate on the amplified negative effects caused by batch dependency and oracle model selection.

2.1.2 Non-trivial Model Selection During TTA

Proper model selection during TTA is crucial for estimating the optimal performance under every choice of hyperparameters. While the model selection is becoming a common practice in Domain Generalization (DG) (Gulrajani & Lopez-Paz, 2021), its importance and necessity in TTA have not been previously acknowledged. The challenges in TTA also differ from those of traditional DG, due to the issues of (i) the lack of validation set and label information during test time; and (ii) batch-dependency issues emerged in the streaming test mini-batches making the oracle model selection method challenging[1].

Observation: oracle model selection is not optimal in TTA. Figure 1(c) (the details of evaluation setups can be seen in appendix D.3) indicates that utilizing an oracle model selection strategy in TTA methods under an online adaptation setup with sufficient adaptation steps initially improves adaptation performance in the first several test batches, compared to Figure 1(a) and 1(b). However, such improvement is short-lived, as the adaptation performance quickly drops in subsequent test batches. It suggests that *the oracle model selection strategy exacerbates the batch dependency problem when considering its use in isolation.* This phenomenon is consistent across various choices of learning rates. Additionally, we find the same problem in TENT and NOTE as shown in Figure 6 of appendix C.3.

Given that no regularization techniques were applied during test time, the failure of oracle model selection may be attributed to the batch dependency and the model's overfitting to test batches it has seen. To further investigate this hypothesis, we systematically reduced the number of adaptation steps for each test batch in Figure 1(c). Our results indicate that such modification mitigates the deterioration and leads to improved adaptation performance from 61.3% to 69.8%. In appendix C.2, we further investigate the effectiveness of two recent regularization techniques originally proposed for non-stationary data, namely Fisher regularizer (Niu et al., 2022b) and stochastically restoring (Wang et al., 2022).

2.1.3 Two Fair Evaluation Protocols

We discuss the trade-offs between online and episodic adaptation. Episodic adaptation with oracle model selection effectively eliminates the impact of batch dependency, resulting in steady but limited improvements. The use of online adaptation empowers a large potential

[1]DG only considers to examine the time-varying scenarios very recently (Yao et al., 2022)

by accumulating historical knowledge. However, it presents a batch dependency challenge, posing model selection during TTA as a min-max equilibrium optimization problem across time and potentially leading to a significant decline in performance.

Evaluation protocols. Here we consider protocols for both episodic and online adaptation.

- **episodic**: episodic TTA *with* oracle model selection (see Algorithm 1).
- **online**: online TTA *without* oracle model selection and grid search[2] the best performance over combinations of learning rates and adaptation steps.

We tune other method-specific hyperparameters separately. We did not introduce the regularization techniques discussed in §2.1.2 as it is orthogonal to our pursuit and only adds unnecessary evaluation complexity.

2.2 The Quality of Pre-trained Model Bottlenecks TTA

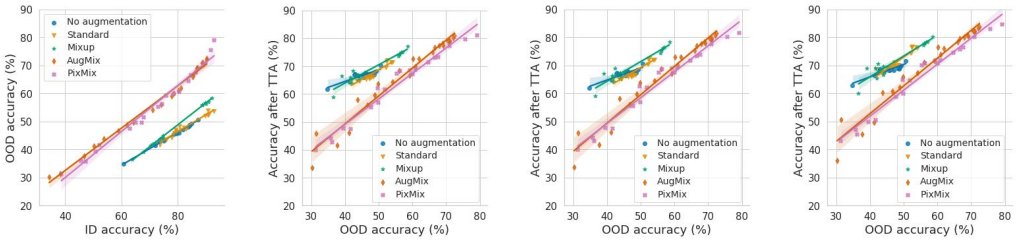
In addition to the unfairness caused by the diverse hyperparameters with improper model selection during TTA, the need of modifying the pre-training phase in Table 4 naturally results in inconsistent model qualities across methods and may deteriorate the test performance even before the TTA. In this section, we conduct a comprehensive and large-scale evaluation to examine the impact of base model quality on TTA performance across various TTA methods. The details of evaluation setups can be seen in appendix D.4.

On the influence of the feature extractor (equivalently full model). The results of our study, as depicted in Figure 8 of appendix G.1, reveal a strong correlation between the performance of test-time augmentation and out-of-distribution generalization on CIFAR10-C. Our analysis shows that across a wide range of TTA methods, *the OOD generalization performance is highly indicative of TTA performance.* A quadratic regression line was found to be an appropriate fit for the data, suggesting that *TTA methods are more effective when applied to models of higher (OOD) quality.*

On the influence of the linear classifier. Our study has revealed that the performance of TTA methods is significantly impacted by the quality of the feature extractor used. The question then arises, can TTA methods bridge the distribution shift gap when equipped with a high-quality feature extractor and a suboptimal linear classifier? Our analysis, as shown in Figure 9(a)-(d) of appendix G.2, indicates that most TTA methods on CIFAR10-C are only able to mitigate the distribution shift gap when the label distribution of the target domain is identical to that of the source domain, at which point the classifier is considered optimal. In this case, TTT attains a 5.2% error rate, the best result observed in test domain #0. However, it is clear that all TTA methods either perform worse than the baseline in the remaining 3 test domains or yield only marginal improvements over the baseline. These findings suggest that *the quality of the classifier plays a crucial role in determining the performance of TTA methods.*

On the influence of the data augmentation strategies. We investigate the impact of various augmentation policies on the performance of ResNet-26 models trained on the CIFAR10 dataset. Our experimental results, as depicted in Figure 2 (more results in Figure 10 of appendix G.3), reveal that models pre-trained with the augmentation techniques like AugMix and PixMix exhibit superior OOD generalization performance on CIFAR10-C compared to models that do not utilize augmentation or only employ standard augmentations. Interestingly, even though *these robust augmentation strategies* significantly improve the robustness of the base model in the target domain, *they only result in a marginal performance increase when combined with TTA.* This disparity is particularly pronounced when compared to the performance of models trained with no augmentation or standard augmentations. However, when all models are fully trained in the source domain, the use of techniques such as AugMix and PixMix still leads to the best adaptation performance on CIFAR10-C, owing to their exceptional OOD generalization capabilities. We reach the same conclusion

[2]Such a traverse, though inefficient, is a clear way to estimate the optimal test performance of each method. We leave the question of optimal model selection in TTA for future research.



(a) ID v.s. OOD (b) OOD v.s. OOD (BN_Adapt) (c) OOD v.s. OOD (TENT) (d) OOD v.s. OOD (SHOT)

Figure 2: **The impact of data augmentation policy on the TTA performance of the target domain.** We save various sequences of checkpoints from the pre-training phase of ResNet-26 with five data augmentation policies and fine-tune each sequence to study the impact of data augmentation. TENT and SHOT use episodic adaptation with oracle model selection. Different data augmentation strategies have different corruption robustness, which causes varying generalization performance on CIFAR10-C. However, good practice in data augmentations and architecture designs for out-of-distribution generalization can be bad for test-time adaptation.

Table 1: **Adaptation performance (error) of TTA methods over OOD datasets with common distribution shifts.** Optimal results in episodic & online are highlighted by **bold** and blue respectively.

	CIFAR10-C (%) ↓	CIFAR10.1 (%) ↓	OfficeHome (%) ↓	PACS (%) ↓	Average error (%) ↓
Baseline	44.3	12.8	39.2	39.5	34.0
BN_Adapt	27.5 ± 0.1	19.0 ± 0.4	39.6 ± 0.1	27.6 ± 0.1	28.4
T3A	40.3 ± 0.1	12.5 ± 0.1	$\mathbf{35.7 \pm 0.1}$	31.0 ± 0.4	29.9
TENT-episodic	26.9 ± 0.0	18.6 ± 0.4	38.4 ± 0.0	26.1 ± 0.1	27.5
TENT-online	21.7 ± 0.1	17.9 ± 0.2	37.6 ± 0.0	22.7 ± 0.2	25.0
SHOT-episodic	21.6 ± 0.0	11.8 ± 0.2	35.9 ± 0.0	$\mathbf{22.0 \pm 0.1}$	**22.8**
SHOT-online	21.0 ± 0.1	14.8 ± 0.0	35.5 ± 0.1	17.8 ± 0.1	22.3
TTT-episodic	$\mathbf{20.9 \pm 0.4}$	12.5 ± 0.1	40.2 ± 0.0	25.3 ± 0.1	24.8
TTT-online	20.0 ± 0.1	13.5 ± 0.0	42.2 ± 0.1	26.6 ± 0.1	25.6
MEMO-episodic	38.1 ± 0.1	$\mathbf{10.8 \pm 0.1}$	37.6 ± 0.0	39.4 ± 0.0	31.5
MEMO-online	85.2 ± 0.7	14.2 ± 1.1	91.3 ± 0.1	75.5 ± 0.4	66.6

across both evaluation protocols and different architectures (e.g., WideResNet40-2) as shown in appendix G.3.

3 Benchmarking Toward Revolutionizing TTA

The efficacy of TTA is contingent upon the nature of distributional variations. Specifically, the advantages demonstrated in previous research in the context of uncorrelated attribute shifts cannot be extrapolated to other forms of distributional shifts, such as shifts in spurious correlation, label shifts, and non-stationary shifts. In this section, we employ two evaluation protocols previously outlined in §2 to re-evaluate commonly used datasets for distributional shifts, as well as benchmarks for distributional shifts that have been infrequently or never evaluated by prevalent TTA methods.
Table 1 and Table 2 summarizes the results of our experiments on all benchmarks for distributional shifts. Details of evaluation setups can be found in appendix D.1.

Common distribution shifts. Here our evaluation of TTA performance primarily focuses on three areas: synthetic co-variate shift (i.e. CIFAR10-C), natural shift (i.e. CIFAR10.1), and domain generalization (i.e. OfficeHome and PACS). *Except for online MEMO, all methods improve average performance across four common distributional shifts*, although the extent of the adaptation performance gain varies among different TTA methods. Notably, online MEMO resulted in a significant degradation in adaptation performance, with an average test error of 66.6%, compared to 31.5% for episodic MEMO and 34.0% for the baseline, indicating that MEMO is only effective in episodic adaptation settings. Additionally, *BN_Adapt,*

Table 2: **Adaptation performance (error in %) of TTA methods over OOD datasets with two realistic distribution shifts.** Dirichlet distribution is used to create non-i.i.d. test streams; the smaller value of α is, the more severe the label shift will be. Optimal results in episodic & online are highlighted by **bold** and blue respectively.

	Spurious correlation shifts ↓		**Label shifts** on CIFAR10 ↓		
	ColoredMNIST	Waterbirds (worst-group)	$\alpha=0.0$	$\alpha=0.1$	$\alpha=1$
Baseline	85.6	29.1	7.8 ± 2.3	5.5 ± 1.3	6.5 ± 0.8
BN_adapt	83.9 ± 0.2	38.1 ± 1.0	77.8 ± 1.7	64.5 ± 7.7	18.2 ± 1.0
T3A	88.1 ± 0.1	**22.3 ± 0.2**	15.9 ± 3.5	9.6 ± 0.7	7.2 ± 0.6
TENT-episodic	83.9 ± 0.2	37.7 ± 1.0	76.8 ± 1.9	63.3 ± 7.1	17.6 ± 0.8
TENT-online	84.3 ± 0.2	24.2 ± 0.4	76.3 ± 2.1	62.2 ± 6.5	16.2 ± 0.4
SHOT-episodic	83.0 ± 0.3	29.4 ± 0.3	10.1 ± 2.5	7.3 ± 1.0	6.6 ± 0.8
SHOT-online	89.7 ± 0.2	27.0 ± 0.7	39.1 ± 3.1	30.0 ± 3.3	10.7 ± 1.0
TTT-episodic	**78.1 ± 0.1**	28.2 ± 0.3	11.0 ± 3.0	5.8 ± 1.7	6.6 ± 1.6
TTT-online	67.1 ± 1.3	24.0 ± 1.9	9.0 ± 2.3	6.1 ± 1.3	7.2 ± 1.4
MEMO-episodic	84.9 ± 0.1	34.3 ± 0.1	**0.1 ± 0.0**	**1.2 ± 0.9**	**4.5 ± 0.6**

TENT, and TTT were unable to ensure improvement in adaptation performance on more challenging and realistic distributional shift benchmarks, such as CIFAR10.1 and OfficeHome. It should be noted that *no single method consistently outperforms the others across all datasets under our fair evaluation.*

Spurious correlation shifts. To the best of our knowledge, this study represents the first examination of the efficacy of dominant TTA methods in addressing spurious correlation shifts as demonstrated in the ColoredMNIST and Waterbirds benchmarks. As shown in Table 2, while some TTA methods demonstrate a reduction in error rate compared to the baseline, *none of TTA methods can improve performance on the ColoredMNIST benchmark*, as even a randomly initialized model exhibits a 50% error rate on this dataset. *In terms of addressing the spurious correlation shift in the Waterbirds dataset, only T3A and TTT can consistently improve adaptation performance*, as measured by worst-group error. TENT and SHOT may potentially improve performance on Waterbirds, but only through the utilization of impractical model selection techniques. The adaptation results presented in appendix H, are obtained through the use of commonly accepted practices in terms of hyperparameter choices, and adhere to the evaluation protocol established in previous research.

Label shifts. Boudiaf et al. (2022) and Gong et al. (2022a) have taken label shift into account in their research, but they paired it with co-variate shift on CIFAR10-C. In contrast, our work solely examines the effectiveness of various TTA methods in addressing label shifts on the CIFAR10 dataset. The experimental results indicate that *all TTA methods, except the MEMO method, demonstrate a higher test error than the baseline under strong label shift conditions.* Specifically, TTA methods that heavily rely on the test batch for recalculating Batch Normalization statistics, such as TENT and BN_Adapt, experience the most significant performance degradation, with BN_Adapt incurring a 77.8% test error and TENT experiencing over 76.0% error rate when the label shift parameter α is set to 0.01.

Non-stationary shifts. In Table 3 we report the adaptation performance of TTA methods on the temporally correlated CIFAR10-C dataset introduced in Gong et al. (2022a). Additionally, we reproduce NOTE in TTAB, which is the current SOTA in the benchmark of temporal correlated shifts. Our results indicate that, even with the appropriate model selection, TENT and BN_Adapt still fail to improve adaptation performance in the presence of non-stationary shifts. However, TTA methods (*e.g.*, TTT and MEMO) demonstrate substantial performance gains when adapting to the temporally correlated test stream, likely due to their instance-aware adaptation strategies, which focus on individual test samples. Surprisingly, MEMO outperforms NOTE in our implementation, which demonstrates the necessity of proper model selection in the field.

Table 3: **Adaptation performance (error in %) of TTA methods on continual distribution shifts.** To make a fair comparison, we employ Batch Normalization (BN) layer and use the same checkpoint with the other methods in NOTE-episodic and NOTE-online. We reproduce the original implementation (with Instand-aware BN) and pretrain another base model in NOTE-online $\star$.

	Cont. dist. shifts (CIFAR10-C)
Baseline	44.3
BN_adapt	79.9 ± 0.5
T3A	43.2 ± 0.3
TENT-episodic	79.2 ± 0.4
TENT-online	79.6 ± 0.4
SHOT-episodic	41.3 ± 0.1
SHOT-online	51.2 ± 2.0
TTT-episodic	27.8 ± 0.1
TTT-online	29.7 ± 0.9
MEMO-episodic	$\mathbf{12.7} \pm 0.1$
NOTE-episodic	39.2 ± 0.1
NOTE-online	25.7 ± 0.1
NOTE-online $\star$	21.8 ± 0.0

4 Conclusion

We have presented TTAB, a large-scale open-sourced benchmark for test-time adaptation. Through thorough and systematic studies, we showed that current TTA methods fall short in three aspects critical for practical applications, namely the difficulty in selecting appropriate hyper-parameters due to batch dependency, significant variability in performance sensitive to the quality of the pre-trained model, and poor efficacy in the face of certain classes of distribution shifts. We hope the proposed benchmark will stimulate more rigorous and measurable progress in future test-time adaptation research.

Acknowledgement

We thank the anonymous reviewers for their constructive and helpful reviews. This work was supported in part by the Science and Technology Innovation 2030 – Major Project (No. 2022ZD0115100), the Research Center for Industries of the Future (RCIF) at Westlake University, Westlake Education Foundation, and the Swiss National Science Foundation under Grant 2OOO21-L92326.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Sara Beery, Yang Liu, Dan Morris, Jim Piavis, Ashish Kapoor, Neel Joshi, Markus Meister, and Pietro Perona. Synthetic examples improve generalization for rare classes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 863–873, 2020.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24, 2011.

Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8344–8353, 2022.

Collin Burns and Jacob Steinhardt. Limitations of post-hoc feature alignment for robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2525–2533, 2021.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305, 2022.

Cian Eastwood, Ian Mason, Chris Williams, and Bernhard Schölkopf. Source-free adaptation to measurement shift via bottom-up feature restoration. In *International Conference on Learning Representations*.

Francois Fleuret et al. Uncertainty reduction for model adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9613–9623, 2021.

Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A Efros. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Jin Gao, Jialing Zhang, Xihui Liu, Trevor Darrell, Evan Shelhamer, and Dequan Wang. Back to the source: Diffusion-driven test-time adaptation. *arXiv preprint arXiv:2207.03442*, 2022.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Robust continual test-time adaptation: Instance-aware bn and prediction-balanced memory. *arXiv preprint arXiv:2208.05117*, 2022b.

Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and Zico Kolter. Test-time adaptation via conjugate pseudo-labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=lQdXeXDoWtI.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021b.

Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. Pixmix: Dreamlike pictures comprehensively improve safety measures. *CVPR*, 2022.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34: 2427–2440, 2021.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Jogendra Nath Kundu, Akshay R Kulkarni, Suvaansh Bhambri, Deepesh Mehta, Shreyas Anand Kulkarni, Varun Jampani, and Venkatesh Babu Radhakrishnan. Balancing discriminability and transferability for source-free domain adaptation. In *International Conference on Machine Learning*, pp. 11710–11728. PMLR, 2022.

Jonghyun Lee, Dahuin Jung, Junho Yim, and Sungroh Yoon. Confidence score for source-free unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 12365–12377. PMLR, 2022.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9641–9650, 2020.

Xianfeng Li, Weijie Chen, Di Xie, Shicai Yang, Peng Yuan, Shiliang Pu, and Yueting Zhuang. A free lunch for unsupervised domain adaptive object detection without source data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8474–8481, 2021a.

Yizhuo Li, Miao Hao, Zonglin Di, Nitesh Bharadwaj Gundavarapu, and Xiaolong Wang. Test-time personalization with a transformer for human pose estimation. *Advances in Neural Information Processing Systems*, 34:2583–2597, 2021b.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pp. 6028–6039, 2020.

Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. TTT++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54 (6):1–35, 2021.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pp. 10–18. PMLR, 2013.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *The Internetional Conference on Machine Learning*, 2022a.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 16888–16905, 2022b.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? 2018. https://arxiv.org/abs/1806.00451.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Evgenia Rusak, Steffen Schneider, George Pachitariu, Luisa Eck, Peter Vincent Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. If your data distribution shifts, use self-learning. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=vqRzLv6POg.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33:11539–11551, 2020.

Samarth Sinha, Peter Gehler, Francesco Locatello, and Bernt Schiele. Test: Test-time self-training under distribution shift. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2759–2769, 2023.

Yongyi Su, Xun Xu, and Kui Jia. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering. In *Advances in Neural Information Processing Systems*, 2022.

Qingyao Sun, Kevin Murphy, Sayna Ebrahimi, and Alexander D'Amour. Beyond invariance: Test-time label-shift adaptation for distributions with" spurious" correlations. *arXiv preprint arXiv:2211.15646*, 2022.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7472–7481, 2018.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7211, 2022.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dj Dvijotham, and Ali Taylan Cemgil. A fine-grained analysis on distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Dl4LetuLdyK.

Shiqi Yang, Yaxing Wang, Kai Wang, Shangling Jui, et al. Attracting and dispersing: A simple approach for source-free domain adaptation. In *Advances in Neural Information Processing Systems*, 2022.

Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems*, 34:23664–23678, 2021.

Marvin Mengxin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. In *Advances in Neural Information Processing Systems*.

Aurick Zhou and Sergey Levine. Bayesian adaptation for covariate shift. *Advances in Neural Information Processing Systems*, 34:914–927, 2021.

Contents of Appendix

A Preliminary and Related Work

Let $\mathcal{D}_{\mathcal{S}} = \{\mathcal{X}_{\mathcal{S}}, \mathcal{Y}_{\mathcal{S}}\}$ be the data from the source domain $\mathcal{S}$ and $\mathcal{D}_{\mathcal{T}} = \{\mathcal{X}_{\mathcal{T}}, \mathcal{Y}_{\mathcal{T}}\}$ be the data from the target domain $\mathcal{T}$ to adapt to. Each sample and the corresponding true label pair $(\mathbf{x}_i, y_i) \in \mathcal{X}_{\mathcal{S}} \times \mathcal{Y}_{\mathcal{S}}$ in the source domain follows a probability distribution $P_{\mathcal{S}}(\mathbf{x}, y)$. Similarly, each test sample from the target domain and the corresponding label at test time t, $(\mathbf{x}^{(t)}, y^{(t)}) \in \mathcal{X}_{\mathcal{T}} \times \mathcal{Y}_{\mathcal{T}}$, follows a probability distribution $P_{\mathcal{T}}(\mathbf{x}, y)$ where $y^{(t)}$ is unknown for the learner. $f_{\boldsymbol{\theta}^o}(\cdot)$ is a base model trained on labeled training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\boldsymbol{\theta}^o$ denotes the base model parameters. During the inference state, there is a drastic performance drop when encountering out-of-distribution test samples, namely $\mathbf{x} \sim P_{\mathcal{T}}(\mathbf{x})$, where $P_{\mathcal{T}}(\mathbf{x}) \neq P_{\mathcal{S}}(\mathbf{x})$.

In contrast to traditional DA that uses $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{X}_{\mathcal{T}}$ collected beforehand for adaptation, TTA adapts a pre-trained model $f_{\boldsymbol{\theta}^o}(\cdot)$ from $\mathcal{D}_{\mathcal{S}}$, only by utilizing test sample $\mathbf{x}^{(t)}$ obtained at test time t. *Episodic* TTA is a stateless TTA variant that always performs adaptation on the base model $\boldsymbol{\theta}^o$, where the adapted model $\boldsymbol{\theta}^\star$ will be discarded after the adaptation step. *Online* TTA instead adapts $\boldsymbol{\theta}^\star$ (obtained from the previous adaptation step) on each new incoming data.

One simple form of TTA is to update the statistics and/or parameters associated with the batch normalization layers (Schneider et al., 2020; Wang et al., 2021). Yet, this design choice is often limited to a narrow set of distribution shifts due to its inherent capacity for adaptation (Burns & Steinhardt, 2021). To effectively update more parameters of deep neural networks, e.g., the feature extractor, from unlabeled test examples, prior works have explored a wide array of proxy objectives. A primary source of such a proxy lies in the common properties of classification problems, e.g., entropy minimization (Liang et al., 2020; Fleuret et al., 2021; Zhou & Levine, 2021), class prototypes (Li et al., 2020; Su

et al., 2022; Yang et al., 2022), pseudo labels (Rusak et al., 2022; Li et al., 2021a), and invariance to augmentation (Zhang et al.; Kundu et al., 2022). However, these techniques are often heavily restricted to the cross-entropy loss of the main classification task, and inherently inapplicable to regression problems, e.g., pose estimation (Li et al., 2021b). Another line of work seeks more general proxies through self-supervised learning, e.g., rotation prediction (Sun et al., 2020), contrastive learning (Liu et al., 2021; Chen et al., 2022), and masked auto-encoder (Gandelsman et al.). Nevertheless, they often share a common downside in that modifying the training process for the auxiliary self-supervised task is necessary but non-trivial. Some other recent works draw inspiration from related problem domains and introduce components to robustify test-time adaptation. Examples include test-time feature alignments (Liu et al., 2021; Eastwood et al.), generative style transformation (Gao et al., 2022), and meta-learning (Zhang et al., 2021), to name a few. Unfortunately, what often comes with a new method being proposed is the skepticism about the efficacy and reproducibility of its precedents (Boudiaf et al., 2022), and yet no prior works have systematically studied the state of the field. Unlike previous efforts introducing new methods, our work makes a step to separate noises from actual progress in the existing ones by establishing a comprehensive benchmark that reflects on the core ideas that stay across a wide range of practical settings and, more importantly, the common pitfalls remaining in the field, to which the future work should pay closer attention.

B Messages

We summarize some key messages of the manuscript here.

Limit 1: unfair evaluation in TTA

- Methods are evaluated under distinct model statuses and experimental setups, e.g.,
  1. model quality used for the adaptation
  2. pretraining procedure
  3. optimizer used for the adaptation
  4. learning rate
  5. # of the adaptation steps per test mini-batch
  6. size of the test min-batch
  7. online v.s. offline adaptation
  8. w/ v.s. w/o resetting model (episodic v.s. online)
- Methodology designs are biased to some specific neural architectures, and TTA methods cannot be fairly evaluated over various neural architectures;

Limit 2: pitfalls of model selection in TTA

- due to the lack of validation set and label information during test time.
- batch-dependency issue emerged in the streaming test mini-batches makes the oracle model selection method challenging[a].

[a] note that the domain generalization field only starts to examine the time-varying scenarios very recently (Yao et al., 2022)

Take-away messages

- Improper evaluation in TTA methods. Hyperparameters have a strong influence on the effectiveness of TTA, and yet they are exceedingly difficult to choose in practice without prior knowledge of the properties and structures of distribution shifts. Even when the labels of test examples are available, selecting the TTA hyperparameters for model selection remains challenging, largely due to batch dependency during online adaptation.
- Batch dependency is a significant issue restricting the performance of online TTA methods. Tackling the batch dependency issue of TTA methods or enabling effective model selection methods is beyond the scope of this manuscript and we leave it to the whole community for future work.
- Pre-trained model quality matters for TTA methods. Even if hyperparameters are optimally selected given oracle information in the test domain, the effectiveness of TTA is not equal on different models. The degree of improvement strongly depends on the quality of the pre-trained model, not only on its accuracy in the source domain but also on its feature properties. Good practice in data augmentations (Hendrycks et al., 2019; 2022) for out-of-distribution generalization leads to reverse effects for TTA.
- The community of TTA needs a comprehensive benchmark such as TTAB to guard effective progress. For example, even under ideal conditions where optimal hyperparameters are used in conjunction with suitable pre-trained models, existing methods still perform poorly on certain classes of distribution shifts, such as correlation shifts (Sagawa et al., 2019) and label shifts (Sun et al., 2022))

Table 4: **Inconsistent evaluation of representative TTA methods.** We only list some key factors due to the space issue; more details refer to ***Limit 1*** in appendix B. The "adjust pre-training" results in distinct models and may dramatically impact the observations (see §2.2).

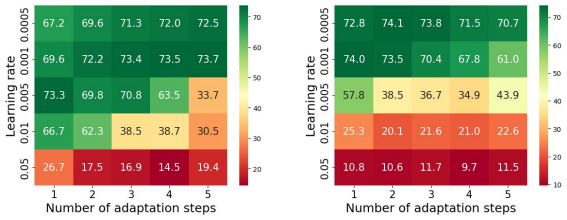
TTA methods	Venue	Adjust pretraining	Access to source domain	Reuse test data	Coupled w/ BatchNorm	Resetting model	Optimizer
BN_Adapt (Schneider et al., 2020)	NeurIPS 2020	✗	✗	✗	✓	✗	-
SHOT (Liang et al., 2020)	ICML 2020	✗	✗	✓	✗	✗	SGD
TTT (Sun et al., 2020)	ICML 2020	✓	✗	✗	✗	✗	SGD
TENT (Wang et al., 2021)	ICLR 2021	✗	✗	✗	✓	✗	Adam & SGDm
T3A (Iwasawa & Matsuo, 2021)	NeurIPS 2021	✗	✗	✗	✗	✗	-
TTT++ (Liu et al., 2021)	NeurIPS 2021	✓	✓	✓	✗	✗	SGDm
TTAC (Su et al., 2022)	NeurIPS 2022	✗	✓	✗	✗	✗	SGDm
Conjugate PL (Goyal et al., 2022)	NeurIPS 2022	✗	✗	✗	✓	✗	Adam
MEMO (Zhang et al.)	NeurIPS 2022	✗	✗	✗	✗	✓	SGD
NOTE (Gong et al., 2022a)	NeurIPS 2022	✓	✗	✗	✓	✗	Adam

C The Limits of Evaluation for TTA Methods

C.1 Improper Evaluation in Prior Work

TTA methods are highly sensitive to the choices of hyperparameters. Selecting hyperparameters for TTA methods is challenging in practice, due to the inaccessible knowledge of the distribution shifts presented in the test streams. Existing evaluation usually leaves this under-explored, either by reusing existing values in the literature regardless of the testing scenario, or tuning in a limited region (see Table 4 for a reference). However, *the effectiveness of TTA methods is heavily dependent on the selection of hyperparameters*: we can witness from a comprehensive evaluation in Figure 3, an improper choice of hyperparameters can lead to a significant degradation in accuracy, with a decrease of up to 59.2% for TENT and 64.4% for SHOT.

Inconsistent adaptation setups across methods further hinder fair evaluation. To study the fairness of evaluation for TTA methods, we outline the key factors that characterize the adaptation procedure of various TTA methods in Table 4. We observe that *even with a narrow range of settings, the experimental configurations for different TTA methods vary greatly.* The negative effect of this inconsistency would be further amplified by the high sensitivity to the hyperparameter configurations, making it difficult to conduct a fair evaluation across TTA methods.

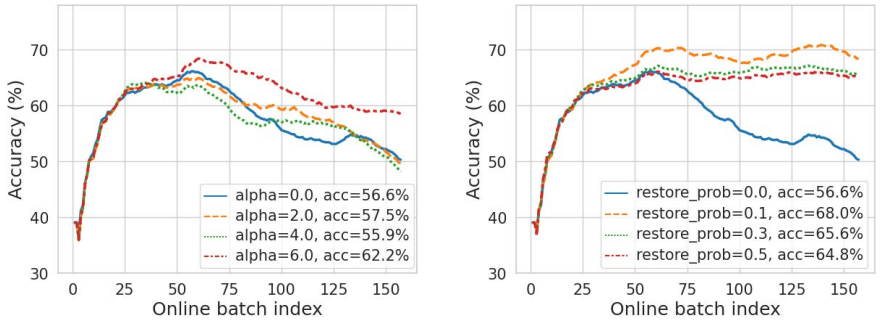


(a) TENT (b) SHOT

Figure 3: **On the hyperparameter sensitivity of TTA methods**, for evaluating the adaptation performance (test accuracy) of TENT and SHOT on CIFAR10-C (gaussian noise), under the combinations of learning rate and # of adaptation steps. This phenomenon occurs in all corruption types.

C.2 Recent Regularization Techniques Proposed to Resist Batch Dependency Problem

We further investigate the effectiveness of two recent regularization techniques originally proposed for non-stationary data, namely Fisher regularizer (Niu et al., 2022b) and stochastically restoring (Wang et al., 2022). Our results in Figure 4 of appendix C.2 indicate that *while these strategies may alleviate the negative effects of batch dependence to some extent, there is currently no principle to trade-off the adaptation and regularization within a test batch, and leave the challenge of balancing adaptation across batches touched.* These techniques are infeasible to consider in model selection and cannot provide a fair assessment for TTA methods, due to the increased sensitivity to their hyperparameters; see a significant variance caused by the regularization method across different learning rates and adaptation steps in Figure 5.



(a) Fisher regularizer (b) Stochastic restoring

Figure 4: On the effect of fisher regularizer and stochastic restoring on batch dependency problem.

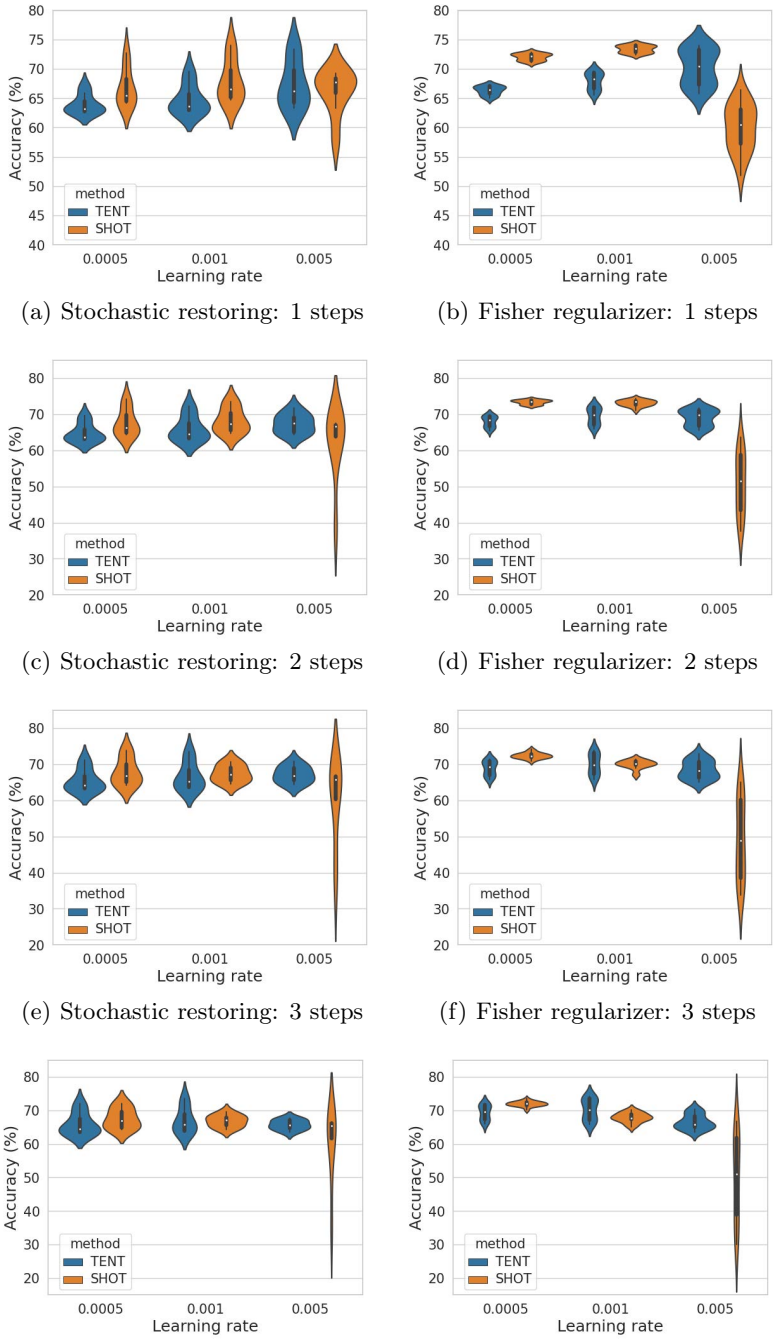
C.3 Optimal Model Selection for TTA is Non-trivial

Oracle model selection protocol also fails to solve the batch dependency issue in TENT and NOTE as shown in Figure 6

D Implementation Details

D.1 Implementation Details of TTA Methods

Following prior work Gulrajani & Lopez-Paz (2021); Sun et al. (2020); Wang et al. (2022), we use ResNet-18/ResNet-26/ResNet-50 as the base model on ColoredMNIST/CIFAR10-C/large-scale image datasets and always choose SGDm as the optimizer. We choose method-specific hyperparameters following prior work. Following Iwasawa & Matsuo (2021), we assign the pseudo label in SHOT if the predictions are over a threshold which is 0.9 in our experiment and utilize $\beta = 0.3$ for all experiments except $\beta = 0.1$ for ColoredMNIST just as Liang et al. (2020). We set the number of augmentations $B = 32$ for small-scale images (e.g. CIFAR10-C, CIFAR100-C) and $B = 64$ for large-scale image sets like ImageNet-C, becasue this is the default option in Sun et al. (2020) and Zhang et al.. We simply set $N = 0$ that controls



(a) Stochastic restoring: 1 steps (b) Fisher regularizer: 1 steps

(c) Stochastic restoring: 2 steps (d) Fisher regularizer: 2 steps

(e) Stochastic restoring: 3 steps (f) Fisher regularizer: 3 steps

(g) Stochastic restoring: 4 steps (h) Fisher regularizer: 4 steps

Figure 5: **The standard deviation of stochastic restoring and Fisher regularizer when considering multiple adaptation steps**. Fisher regularizer (Niu et al., 2022b) aims to constrain important model parameters from drastic changes to alleviate the error accumulated due to batch dependency. Stochastically restoring (Wang et al., 2022) involves a small portion of model parameters to their pre-trained values after adaptation on each test batch to prevent catastrophic forgetting. The hyperparameter tuning for these two techniques is challenging due to the high degree of variability inherent in these methods, which might impede their practical utility, particularly when compounded by the issue of batch dependency.

the trade-off between source and estimated target statistics because it achieves performance comparable to the best performance when using a batch size of 64 according to Schneider

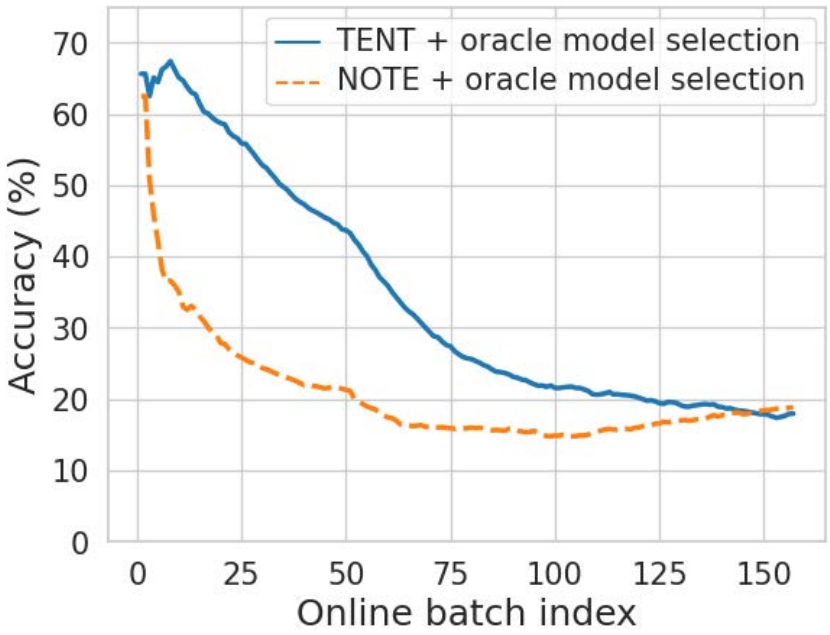


Figure 6: Oracle model selection also fails in TENT and NOTE under the online setting. Here we use ResNet-26 as the base model and learning rate is equal to 0.005.

et al. (2020). Training-domain validation data is used to determine the number of supports to store in T3A following Iwasawa & Matsuo (2021). We keep the average performance on the dataset if it has multiple test domains (e.g., CIFAR10-C, OfficeHome) and calculate the standard deviation over three different trials {2022, 2023, 2024}. We always examine the highest severity of corrupted data throughout our study.

D.2 Implementation Details of TTAB Methods

To establish a consistent and realistic evaluation framework for TTA methods, we have implemented several key choices. ① In contrast to the inconsistent pre-training strategies employed in previous studies, we have adopted a self-supervised learning approach utilizing the rotation prediction task as an auxiliary head, in conjunction with standard data augmentation techniques. This allows us to include TTT variants and maintain a consistent level of model quality across different TTA methods. ② For TTA methods that adapt a single image at a time (such as MEMO and TTT), we have modified the optimization procedure to accommodate larger batch sizes. Specifically, we have fixed the model parameters and accumulated gradients computed for each sample in a batch, only updating the model parameters once all samples have been adapted in a batch. Such a design excludes the unfairness caused by varied mini-batch sizes. ③ We have utilized Stochastic Gradient Descent with momentum for TTA throughout all experiments conducted in this work (see the discrepancy in Table 4).

D.3 Evluation setups for oracle model selection

To support our claims, we propose using an oracle-based model selection method specifically designed for TTA under the online adaptation setting. We assume access to true labels and select the optimal model (with early stopping) for each test batch with a sufficient number of adaptation steps. This approach is expected to achieve the highest possible adaptation performance per adaptation batch. The implementation is detailed in Algorithm 1.

For the sake of simplicity, we select the method-specific hyperparameters of each TTA method following the prior work (see more details in appendix D.1), while focusing on tuning two key adaptation-specific hyperparameters, namely learning rate and number of

Algorithm 1 Oracle model selection for online TTA

Input: model state $\boldsymbol{\theta}^o$, test sample $\mathbf{x}^{(t)}$, true label $y^{(t)}$, maximum adaptation steps M, learning rate η, objective function ℓ, update rule $\mathcal{G}$, and model selection metric $\mathcal{J}$.
procedure ORACLE_MODEL_SELECTION($\boldsymbol{\theta}, \ldots$)
  Initialize: $m \leftarrow 1, \mathcal{F} \leftarrow \{\boldsymbol{\theta}\}, \boldsymbol{\theta}_m \leftarrow \boldsymbol{\theta}$
  for $m \in \{1, \cdots, M\}$ **do**
    Compute loss $\tilde{\ell} \approx \ell(\boldsymbol{\theta}_m; x^{(t)})$
    Adapt parameters via $\boldsymbol{\theta}_{m+1} \leftarrow \mathcal{G}(\boldsymbol{\theta}_m, \eta, \tilde{\ell})$
    $\mathcal{F} \leftarrow \mathcal{F} \cup \boldsymbol{\theta}_{m+1}$
  Select optimal model $\boldsymbol{\theta}^\star \leftarrow \arg\max_{\tilde{\boldsymbol{\theta}} \in \mathcal{F}} \mathcal{J}(\tilde{\boldsymbol{\theta}}, y^{(t)})$
  return Pass $\boldsymbol{\theta}^\star$ to next test sample $\mathbf{x}^{(t+1)}$

(a) formulation. (b) an example of $\mathcal{P}(\mathbf{a}^{1:K})$.

Figure 7: **A formulation of test-time distribution shifts**.

adaptation steps, which are highly relevant to the adaptation process detailed in **??**. We choose the maximum steps in Algorithm 1 as 50 according to our observation in Figure 3 and set the maximum steps as 25 in large-scale datasets due to the computational feasibility.

D.4 Evaluation setups for investigation of pre-trained model quality

We thoroughly examine the pre-trained model quality from the aspects of (1) disentangled feature extractor and classifier, and (2) data augmentation.

1. We consider a model with decoupled feature extractor and classifier. We keep the checkpoints with varying performance levels, generated from the pre-training phase using the standard data augmentation technique (mentioned below). We then fine-tune a trainable linear classifier for each frozen feature extractor from the checkpoints, using data with a uniform label distribution, to study the effect of the feature extractors (equivalently full model). To study the effect of the linear classifiers, we freeze a well-trained feature extractor and fine-tune trainable linear classifiers on several non-i.i.d. datasets created from a Dirichlet distribution; we further use Dirichlet distribution to create non-i.i.d. test data streams.
2. We consider 5 data augmentation policies: (i) no augmentations, (ii) standard augmentation, i.e. random crops and horizontal flips, (iii) MixUp (Zhang et al., 2017) combined with standard augmentations, (iv) AugMix (Hendrycks et al., 2019), and (v) PixMix (Hendrycks et al., 2022). For each data augmentation method, we save the checkpoints from the standard supervised pre-training phase to cover a wide range of pre-trained model qualities.

E TTAB: A PyTorch Testbed for Test-Time Adaptation Benchmark

TTAB, our PyTorch testbed for reproducible and rigorous TTA research, lies at the center of our large-scale benchmarking. The initial release comprises 9 algorithms, 6 distribution shift benchmarks, and 2 proper evaluation protocols (those described in §2), as well as the infrastructure to run all the experiments. The current version of TTAB focuses on image classification, leaving for future work other types of tasks.

E.1 A Fine-Grained Formulation of Distribution Shifts

TTAB is built on top of a fine-grained formulation of distribution shifts, allowing great flexibility and extensibility for arbitrary test scenarios in future TTA research. Similar

to Wiles et al. (2022), here we generalize notations in **??** and decompose data into an underlying set of factors of variations: we assume a joint distribution $\mathcal{P}$ of (i) inputs $\mathbf{x}$ and (ii) corresponding attributes $\mathbf{a}^{1:K} := \{\mathbf{a}^1, \ldots, \mathbf{a}^K\}$, where $\mathbf{a}^k$ is sampled from a finite set.

A fine-grained formulation for test-time distribution shifts is formulated in Figure 7. Figure 7(a) decipts an overall framework, by considering ① the underlying distribution of attribute values $\mathcal{P}(\mathbf{a}^{1:K})$, ② sampling operators (e.g., # of sampling trials and sampling distribution), and ③ the concatenation of sampled data over time-slots. Figure 7(b) further elaborates a potential shift caused by three attributes, covering most of the realistic cases illustrated below:

1. attribute-relationship drift (a.k.a. spurious correlation): attributes are correlated under $\mathcal{P}_\mathcal{S}$ but not $\mathcal{P}_\mathcal{T}$.
2. attribute-values drift—the distribution of attribute values under $\mathcal{P}_\mathcal{S}$ are differ from that of $\mathcal{P}_\mathcal{T}$. Its extreme case generalizes to the shift that some attribute values are unseen under $\mathcal{P}_\mathcal{S}$ but are under $\mathcal{P}_\mathcal{T}$.

E.2 Benchmark Overview

Datasets. TTAB includes data loaders for 6 standard benchmarks of distribution shifts. These are CIFAR10-C, CIFAR10.1, OfficeHome, PACS, ColoredMNIST, and Waterbirds. More datasets details can be found in appendix F. It is noteworthy that the design in appendix E.1 enables TTAB to be easily extended to other datasets and test scenarios.

Algorithms. TTAB currently has 9 TTA methods: Test-time Normalization (BN_Adapt (Schneider et al., 2020)), Test Entropy Minimization (TENT (Wang et al., 2021)), Test-time Template Adjuster (T3A (Iwasawa & Matsuo, 2021)), Source Hypothesis Transfer (SHOT (Liang et al., 2020)), Test-time Training (TTT (Sun et al., 2020)), Marginal Entropy Minimization (MEMO (Zhang et al.)), NOn-i.i.d. TEst-time adaptation (NOTE (Gong et al., 2022a)), Fisher Regularizer (Niu et al., 2022a), and stochastically restoring model parameters (Wang et al., 2022). The modular design of the TTAB codebase allows seamless integration of the algorithm with other components (e.g. test dataset, test scenarios, and model selection). Other implementation details of TTAB can be found in appendix D.2.

F Datasets

TTAB includes downloaders and loaders for all image classification tasks considered in our work:

- **ColoredMNIST** Arjovsky et al. (2019) is a variant of the MNIST handwritten digit classification dataset. Domain $d \in \{0.1, 0.3, 0.9\}$ contains a disjoint set of digits colored either red or blue. The label is a noisy function of the digit and color, such that color bears correlation d with the label and the digit bears correlation 0.75 with the label. This dataset contains 70000 examples of dimension $(2, 28, 28)$ and 2 classes.
- **OfficeHome** Venkateswara et al. (2017) comprises four domains $d \in \{$ art, clipart, product, real $\}$. This dataset contains 15,588 examples of dimension $(3, 224, 224)$ and 65 classes.
- **PACS** Li et al. (2017) comprises four domains $d \in \{$ art, cartoons, photos, sketches $\}$. This dataset contains 9,991 examples of dimension $(3, 224, 224)$ and 65 classes.
- **CIFAR10** Krizhevsky et al. (2009) consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.
- **CIFAR10-C** Hendrycks & Dietterich (2019) is a dataset generated by adding 15 common corruptions + 4 extra corruptions to the test images in the Cifar10 dataset.
- **CIFAR10.1** Recht et al. (2018) contains roughly 2,000 new test images that were sampled after multiple years of research on the original CIFAR-10 dataset. The data

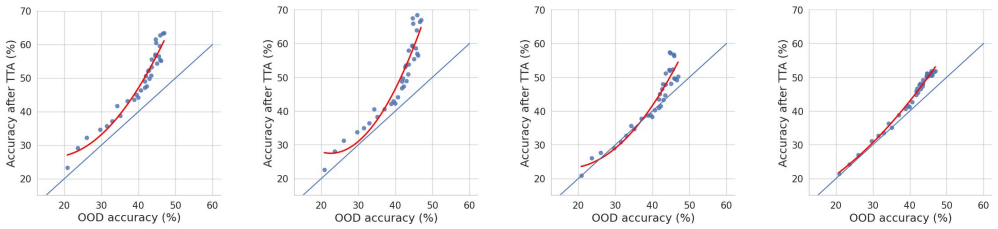
collection for CIFAR-10.1 was designed to minimize distribution shift relative to the original dataset.

- **Waterbirds** Sagawa et al. (2019) is constructed by cropping out birds from photos in the Caltech-UCSD Birds-200-2011 (CUB) dataset and transferring them onto backgrounds from the Places dataset.

G Model Quality

G.1 On the influence of feature extractor

In Figure 8, we show the results of investigating the influence of linear classifier on TTA.



(a) BN_Adapt. (b) TENT. (c) SHOT. (d) T3A.

Figure 8: **The impact of model quality on TTA performance, in terms of OOD v.s. OOD (TTA)** on CIFAR10-C. We save the checkpoints from the pre-training phase of ResNet-26 with standard augmentation. The OOD generalization performance has a significant impact on the overall performance (i.e. averaged accuracy of all corruption types) of various TTA methods. Our analysis reveals a strong correlation between model quality and the effectiveness of TTA methods. Furthermore, certain TTA methods, specifically SHOT, may not provide an improvement in performance on OOD datasets and may even result in a decrease in performance when applied to models of low quality.

G.2 On the influence of linear classifier

In Figure 9, we show the results of investigating the influence of linear classifier on TTA.

G.3 On the Influence of Data Augmentation

In Figure 10, Figure 11, and Figure 12, we show more data augmentation results across different model architectures and different evaluation protocols.

H Additional Results

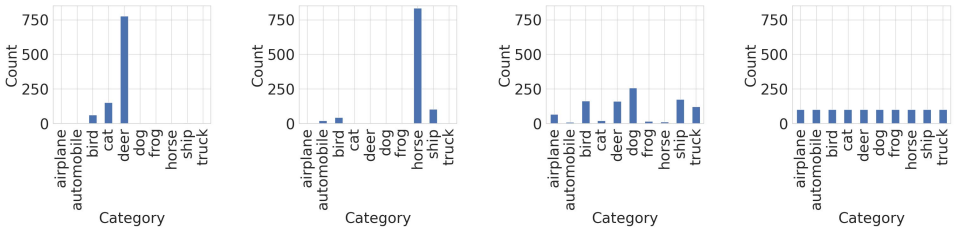
In Table 5, we observe that for each TTA method, if following the evaluation setting in prior work, only T3A and TTT can enhance adaptation performance in terms of worst-group error.

Table 5: TTA performance on Waterbirds using the common practice of hyperparameter choices and following the main setting in prior work.

Methods	Test average error ↓	Test worst-group error ↓
Baseline (ResNet50-JT)	2.22	29.1
BN_Adapt	2.73 ± 0.1	38.0 ± 0.2
T3A	2.27 ± 0.1	21.5 ± 0.2
TENT-online	2.69 ± 0.2	34.1 ± 0.3
SHOT-online	7.82 ± 0.2	42.8 ± 0.5
TTT-online	5.0 ± 0.1	23.0 ± 0.2
MEMO-episodic	3.67 ± 0.1	45.0 ± 0.1

I Additional Related Work

Unsupervised Domain Adaptation Unsupervised Domain Adaptation (UDA) is a technique aimed at enhancing the performance of a target model in scenarios where there is a shift in distribution between the labeled source domain and the unlabeled target domain.



(a) domain #0 (b) domain #1 (c) domain #2 (d) domain #3

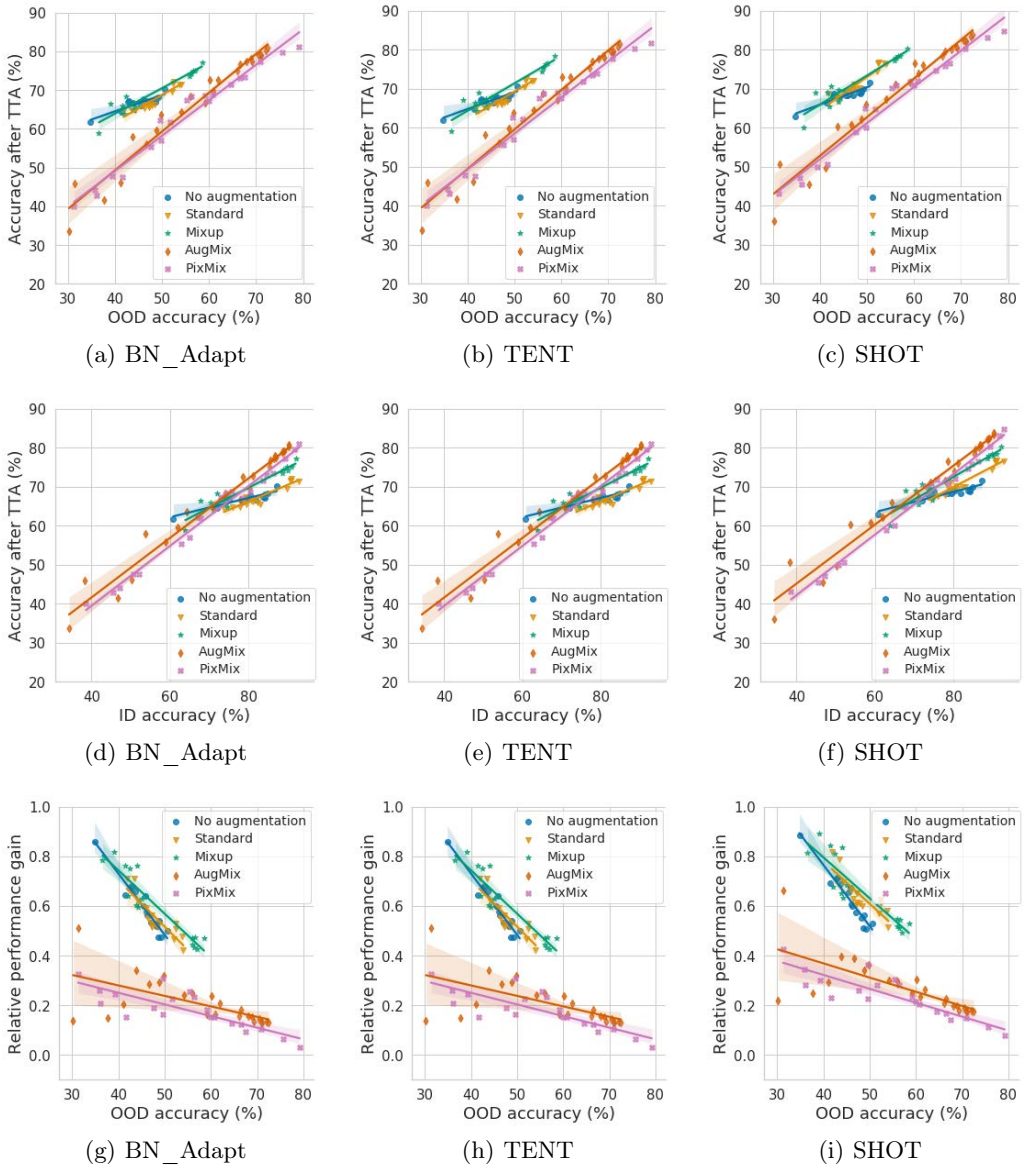
	domain #0 (%) ↓	domain #1 (%) ↓	domain #2 (%) ↓	domain #3 (%) ↓
Baseline	12.6	96.5	79.9	76.1
BN_Adapt	6.6	98.9	88.2	81.4
T3A	28.3	**95.1**	**74.1**	72.4
TENT	6.6	98.9	88.3	81.6
SHOT	6.3	96.5	77.4	73.1
TTT	**5.2**	96.5	77.0	**70.8**
MEMO	10.0	97.9	85.1	80.0

Figure 9: **Adaptation performance (error) of TTA methods over CIFAR10-C with different label shifts**. (a) test domain #0: $\alpha = 0.1$, same label distribution with training environment. (b) test domain #1: $\alpha = 0.1$, different label distribution with training environment. (c) test domain #2: $\alpha = 1$. (d) test domain #3: uniformly distributed test stream. We investigate the impact of the degree of non-i.i.d.-ness in the fine-tuning dataset on the performance of the linear classifier. Our findings reveal that the quality of the linear classifier plays a crucial role in determining the effectiveness of TTA methods, as they can only enhance performance on test data that shares similar i.i.d.-ness and label distribution characteristics. Despite utilizing a well-trained feature extractor, the quality of the linear classifier remains a significant determining factor in the overall performance of TTA methods.

UDA methods typically seek to align the feature distributions between the two domains through the utilization of discrepancy losses (Long et al., 2015) or adversarial training (Ganin & Lempitsky, 2015; Tsai et al., 2018).

Domain Generalization Our work is also related to DG (Muandet et al., 2013; Blanchard et al., 2011) in a broad sense, due to the shared goal of bridging the gap of distribution shifts between the source domain and the target domain. Also, DG and TTA may share similar constraints on model selection for lacking label information in the target domain. DomainBed Gulrajani & Lopez-Paz (2021) highlights the necessity of considering model selection criterion in DG and concludes that ERM (Vapnik, 1998) outperforms the state-of-the-art in terms of average performance after carefully tuning using model selection criteria.

Distribution Shift Benchmarks. Distribution shift has been widely studied in the machine learning community. Prior works have covered a wide range of distribution shifts. The first line of such benchmarks applies different transformations to object recognition datasets to induce distribution shifts. These benchmarks include: (1) CIFAR10-C & ImageNet-C (Hendrycks & Dietterich, 2019), ImageNet-A (Hendrycks et al., 2021b), ImageNet-R (Hendrycks et al., 2021a), ImageNet-V2 (Recht et al., 2019), and many others; (2) ColoredMNIST (Arjovsky et al., 2019), which makes the color of digits a confounder. Most recent benchmarks collect sets of images with various styles and backgrounds, such as PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017), DomainNet (Peng et al., 2019), and Waterbirds (Sagawa et al., 2019). Unlike most prior works that assume a specific stationary target domain, the study on continuous TTA that considers continually changing target data becomes more and more popular in the field. Recently, a few works have constructed datasets and benchmarks for scenarios under temporal shifts. Gong et al. (2022b) builds a temporally correlated test stream on CIFAR10-C sample by a Dirichlet distribution, where most existing TTA methods fail dramatically. Wild-Time (Yao et al., 2022) benchmark consists of 5 datasets that reflect temporal distribution shifts arising in a variety of real-world applications, including patient prognosis and news classification. Studies on fairness and



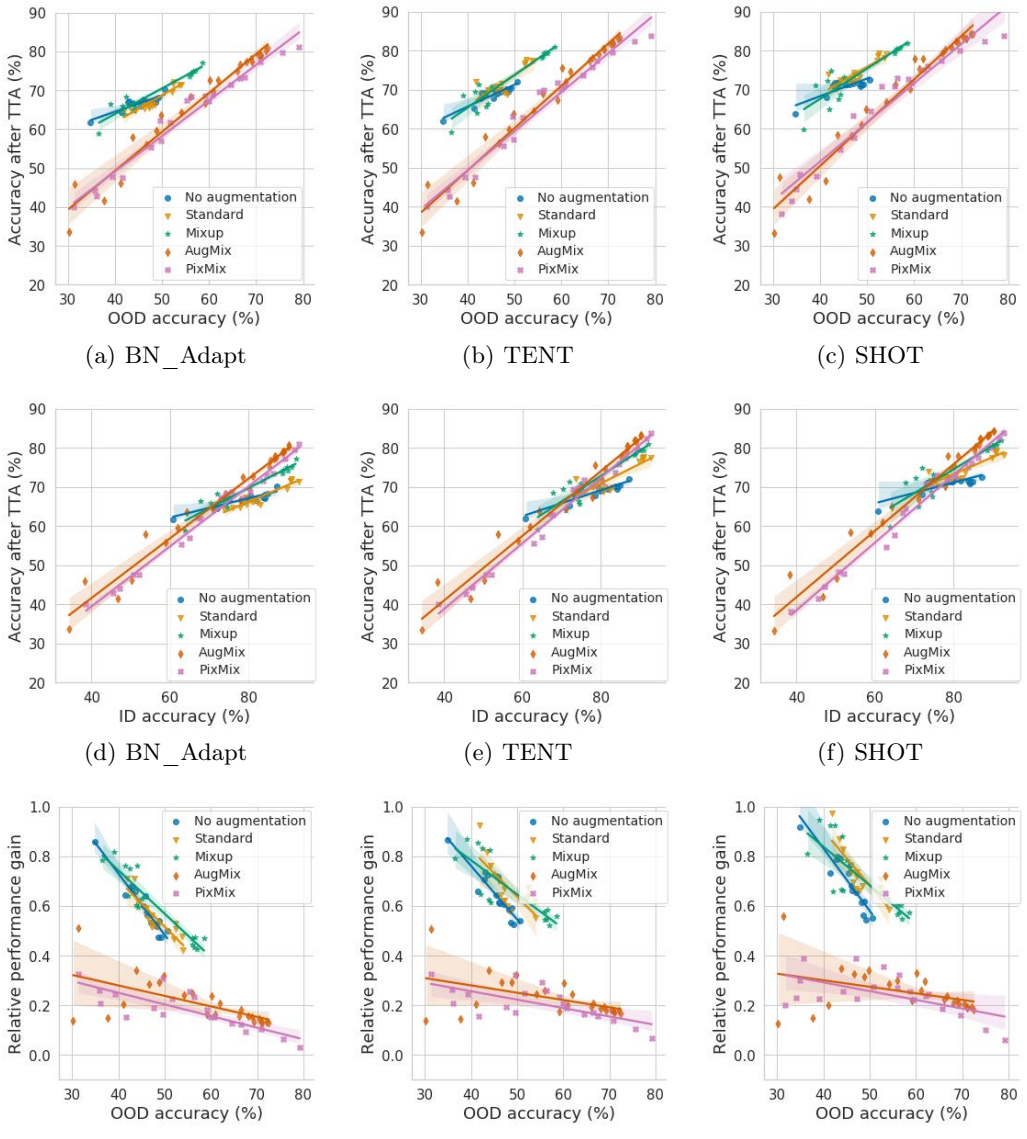
(a) BN_Adapt (b) TENT (c) SHOT

(d) BN_Adapt (e) TENT (f) SHOT

(g) BN_Adapt (h) TENT (i) SHOT

Figure 10: **The effect of data augmentation on TTA performance in the target domain**. TENT and SHOT use episodic adaptation with oracle model selection and choose ResNet-26 as the base model.

bias (Mehrabi et al., 2021) have investigated the detrimental impact of spurious correlation in classification (Geirhos et al., 2018) and conservation (Beery et al., 2020). To our knowledge, there have been rare TTA work focused on tackling spurious correlation shifts.

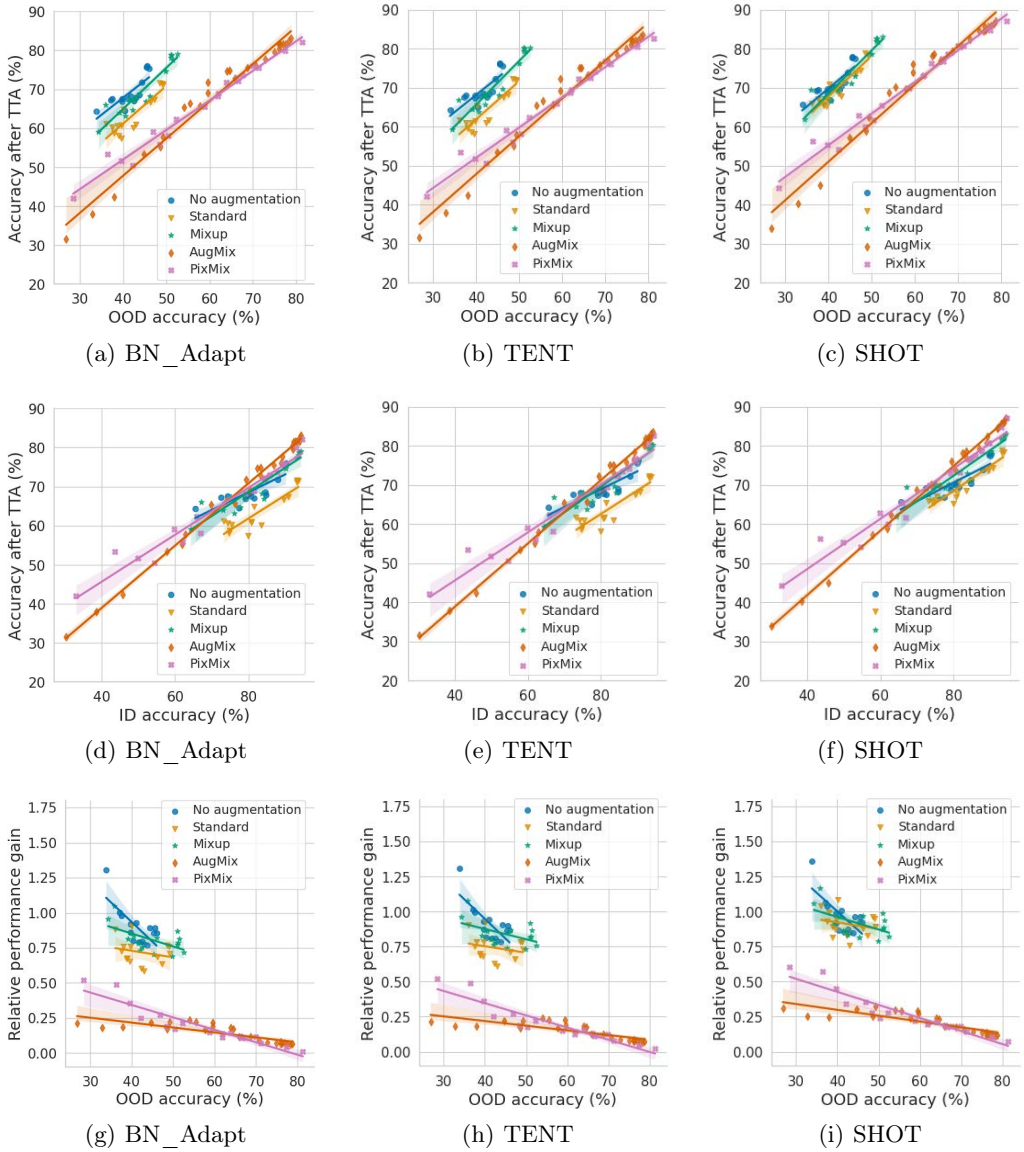


(a) BN_Adapt (b) TENT (c) SHOT

(d) BN_Adapt (e) TENT (f) SHOT

(g) BN_Adapt (h) TENT (i) SHOT

Figure 11: **The effect of data augmentation on TTA performance in the target domain**. TENT and SHOT use online adaptation without oracle model selection and grid search the best performance. We use ResNet-26 as the base model here.



(a) BN_Adapt (b) TENT (c) SHOT

(d) BN_Adapt (e) TENT (f) SHOT

(g) BN_Adapt (h) TENT (i) SHOT

Figure 12: **The effect of data augmentation on TTA performance in the target domain**. TENT and SHOT use episodic adaptation with oracle model selection and choose WideResNet40-2 as the base model.