Channel Selection for Test-Time Adaptation Under Distribution Shift

Pedro Vianna
Université de Montréal

Muawiz Chaudhary
Concordia University
Mila – Quebec AI Institute

An Tang
Université de Montréal

Guy Cloutier
Université de Montréal

Guy Wolf
Université de Montréal
Mila – Quebec AI Institute

Michael Eickenberg
Flatiron Institute

Eugene Belilovsky
Concordia University
Mila – Quebec AI Institute

Abstract

To ensure robustness and generalization to real-world scenarios, test-time adaptation has been recently studied as an approach to adjust models to a new data distribution during inference. Test-time batch normalization is a simple and popular method that achieved compelling performance on domain shift benchmarks by recalculating batch normalization statistics on test batches. However, in many practical applications this technique is vulnerable to label distribution shifts. We propose to tackle this challenge by only selectively adapting channels in a deep network, minimizing drastic adaptation that is sensitive to label shifts. We find that adapted models significantly improve the performance compared to the baseline models and counteract unknown label shifts.

1 Introduction

A commonly cited limitation of deep learning models is the inability to generalize across different domains [1]. Generalization can be simply defined as the ability of an algorithm to be applied to a different, yet still related, target domain. Typically, in real-world deployment scenarios models might encounter data with critical differences, hampering their performance. This decrease in performance has been observed in multiple areas, including life-threatening contexts, such as autonomous driving [2, 3] and medical diagnostics [4, 5].

A recently emerging technique to deal with distribution shift is test-time adaptation (TTA) [6, 7], a type of unsupervised domain adaptation, where unlabeled test data is used to update the model parameters at test-time, before predictions. It is often assumed that data arrives in batches, and some studies have proposed a setting of test-time batch adaptation that take advantage of batch-level information to adapt to the distribution shift [8, 9, 10].

Test-time batch normalization (TTN) [11, 12] replaces batch normalization statistics estimated as running averages on the training set with the statistics of the test data batch. Despite being a simple approach, it has been shown to improve robustness under covariate shift, handling particularly well various cases of image corruptions. Based on that, other TTA approaches apply TTN as a critical component in their foundation [10, 13]. Alas, most existing TTA methods consider the impact of covariate shifts only, in many realistic scenarios the label distribution of data can shift from training to testing.

In this work, we investigate the effect of label distribution shift on TTN and observe that it can lead to catastrophic failures. Moreover, we notice the effects of adapting different layers in TTN. Motivated by it, we propose a method to correct for the label distribution shift based on the adaptation of some channels of the batch normalization layers. Our proposed method is applied for classification of

Workshop on Distribution Shifts, 37th Conference on Neural Information Processing Systems (NeurIPS 2023).

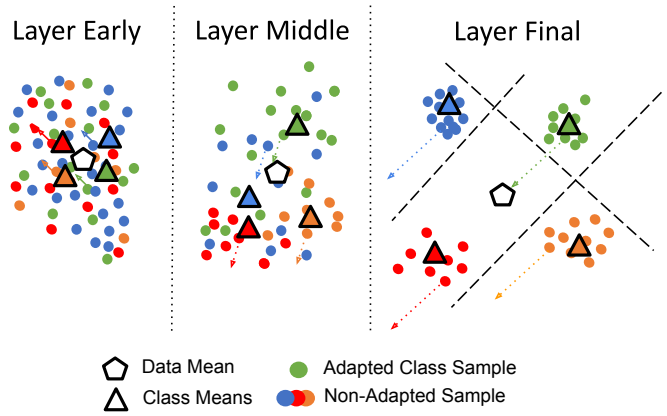

Figure 1: We illustrate a mechanism for explaining the observed behaviour under label distribution shift. We consider one class mean (green) which is shifted towards the data mean, as would be the case in a highly imbalanced setting. Classes are not well separated in early layers and thus shifts in any mean are relatively small and non-intrusive. In later layers classes are well separated and a large shift of points from one mean towards the data mean is likely to cross a decision boundary. Data points in other classes moving away from the data mean are less likely to cross a decision boundary.

two well-known benchmark natural images datasets (CIFAR-10 [14] and ImageNet-1K [15]). When deployed in target data with different distribution, our proposed method is effective for imbalanced adaptation.

2 Hybrid test-time batch normalization

The main idea behind TTA in general, and TTN in particular, is that while label information is not available at test time, the unlabeled data can provide information to estimate impact of domain shifts on neural networks. The typical setup is based on data being processed in batches, enabling assessment of distribution shifts between source and target domains.

In order to implement TTA in these settings, TTN views a neural network f as split into blocks separated by BatchNorm layers:

$$f = f_K \circ B_{K-1} \circ f_{K-1} \circ \cdots \circ B_1 \circ f_1 \circ B_0 \circ f_0, \tag{1}$$

where $f_0, \ldots, f_K$ are blocks (i.e., sub-networks) of hidden layers and $B_0, \ldots, B_{K-1}$ are batch normalization operators, each BatchNorm layer modifies each neural activation by

$$B(h(x), \mu_k, \sigma_k, \beta, \gamma) = \beta \frac{h(x) - \mu_k}{\sigma_k} + \gamma \tag{2}$$

where β and γ are parameters learned during the training process, and μ and σ represent estimates of the mean and standard deviation of neuron activation over data.

The main premise of the TTN approach is that changes in the distributions of activations of each neuron between source and target batches would predominantly be caused by unwanted covariate shifts, and therefore should be eliminated. However, this does not take into account other distribution shifts that *should* affect the output distribution of the network. It is often the case that the distribution of available labels during the training process will differ from one of unknown labels encountered at test time. Most successful applications of TTN did not contain such label distribution shift, and recent work has indicated possible sensitivity of TTN to such shifts [16].

In order to mitigate the risk of adverse effects by TTN, we consider an approach that aims to only adapt channels or neurons which are sensitive primarily to covariate shift, excluding channels which are highly sensitive to shifts in the label distribution. Specifically, consider a model with K layers, for a layer k with source statistics μ_k^s, σ_k^s and target data statistics μ_k^t, σ_k^t, computed for each layer using the input batch. We construct a new set of hybrid statistics, $\mu_k^{hybrid} = m_k \odot \mu_k^t + (1 - m_k) \odot \mu_k^s$ and $\sigma_k^{hybrid} = m_k \odot \sigma_k^t + (1 - m_k) \odot \sigma_k^s$, where the binary mask m_k will aim to not adapt neurons or channels which are highly sensitive to label shifts.

We base our selection of the m_k on two principles, (a) channels in later layers in neural networks are more specialized than earlier layers, which perform generic feature extraction [17, 18, 19] (b) channels in layers which experience largest shifts will tend to be most sensitive to label shift. This intuition is illustrated in Figure 1. We propose to combine these notions as follows: in each layer the top $T\%$ most changed channels as measured by a metric (e.g. Wasserstein distance) will not be updated, limiting the most severe changes. The number of channels to adapt is modulated by $c(i)$ where i is the layer. Based on the notion that later layers should change minimally, the $T\%$ of channels that are not updated will increase with depth. For the rest of the work we will compute distribution shift using the Wasserstein distance between two gaussians, $W_2^2(\{\mu^s, \sigma^s\}, \{\mu^t, \sigma^t\}) = \|\mu^s - \mu^t\|^2 + \sigma^s + \sigma^t - 2\sigma^s\sigma^t$, and for the increase of $T\%$ over layers we use a linear ramp $c(i) = \frac{i}{K}$. The proposed Hybrid-TTN algorithm is described in the Appendix A.1.

3 Experiments and results

We use two popular benchmarks datasets in our evaluations: CIFAR-10-C and ImageNet-1K-C.

CIFAR-10 and CIFAR-10-C. We use the CIFAR-10 [20] dataset along with CIFAR-10-C [21]. CIFAR-10 is a small natural image dataset with 50k training images and 10k validation images. CIFAR-10-C contains corrupted versions of the CIFAR-10 Validation set at varying severities. We train our models on the uncorrupted dataset.

ImageNet-1K and ImageNet-1K-C. We use the ImageNet-1K [22] dataset along with ImageNet-1K-C [21]. ImageNet-1K is a large natural image dataset with 1.2 million training images and 50k validation images. ImageNet-1K-C, similarly to CIFAR-10-C, contains corrupted versions of the ImageNet-1K validation set at varying severities. Both CIFAR-10-C and ImageNet-1K-C are popular as a measure of robustness to covariate shift.

Training and architecture details. On CIFAR-10 we train a Resnet-26 model as defined in [23]. We use an SGD optimizer with a batch size of 128. An initial learning rate set to 0.1 is used in combination with a cosine annealing schedule [24] trained over 200 epochs. Weight decay set to 5e-4 is used along with momentum set to 0.9 [25]. Standard augmentation uses random crop of size 32 with 4 padding, and random horizontal flips. For ImageNet-1K we use a pre-trained Resnet18 model.

Adaptation details. We focus on the TTA setting where adaptation is done on a single batch without affecting the deployed model. We use a batch size of 500 for the experiments (sampled over multiple seeds). For the Hybrid-TTN, we use the Wasserstein distance as the metric for measuring the changes in the adapted channels.

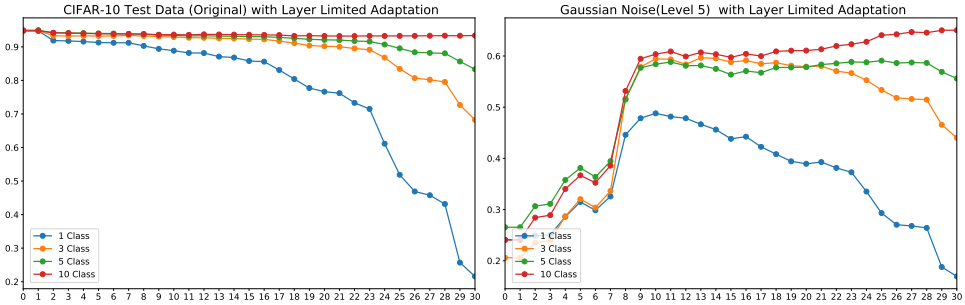

Figure 2: On CIFAR-10 we adapt models only up to the layer shown on the x-axis, the y-axis showing the accuracy on the target data. We consider target data with and without corruptions, and for each we test with different label distributions. We consider label distributions with all (10) classes as well as 5,3, and 1 randomly selected and balanced classes. Note the x-axis starting value is the source model performance and the ending value the TTN model performance. We observe that adapting some layers can avoid the catastrophic collapse due to TTN observed on original data while maintaining the benefits of TTN over the source model in covariate shift.

3.1 Shortcomings of TTN

We first illustrate the potential pitfalls of TTN. Using the CIFAR-10 dataset, we show the effect of label distribution shift on TTN. Moreover, we perform experiments on layer-limited adaptation both with and without noise. Our results are shown in in Figure 2. First, we observe that on the non-corrupted train data the performance of class-imbalanced data degrades gradually at first and increasingly faster towards the later layers. This suggests that later layers can cause a large

degradation. Secondly, for corrupted data we observe that adapting up to earlier layers can allow enough label distribution invariance to provide benefits under covariate shift.

3.2 Evaluating Hybrid-TTN

We now use the proposed Hybrid-TTN method on a variety of target datasets, covariate shifts, and label distribution shifts. We demonstrate that Hybrid-TTN can provide a good trade-off in being able to adapt to covariate shift without experiencing catastrophic failure due to label distribution shift. Our results are shown in Tables 1 and 2. Here we demonstrate various degrees of covariate and label distribution shift, and the gain or loss as compared to the source model performance. Unlike TTN, it is able to handle the label distribution shift, in many cases avoiding catastrophic failure, and in a variety of combinations of severe label and covariate shift improving over the source model. An ablation study, presented in Appendix A.2, indicate that the random selection of channels does not yield good outcomes, validating our premise that the selection of channels with the lowest Wasserstein distance is an effective strategy.

		Covariate Shift					
		Accuracy (% or Δ%)					
Label Distribution Shift		Original	Corruption-1	Corruption-2	Corruption-3	Corruption-4	Corruption-5
Original (10 classes)	Source	94.6 ± 0.8	87.3 ± 9.8	81.7 ± 11.9	74.5 ± 14.8	66.2 ± 17.9	53.8 ± 20.6
	TTN	-1.1 ± 0.6	+2.2 ± 5.7	+5.7 ± 7.5	+10.6 ± 9.7	+16.9 ± 13.4	+24.5 ± 17.7
	Hybrid-TTN	-1.0 ± 0.2	+1.1 ± 3.0	+3.9 ± 4.5	+7.6 ± 7.3	+12.4 ± 10.5	+17.9 ± 15.7
1 class	Source	94.5 ± 3.3	87.2 ± 10.9	80.8 ± 14.7	74.7 ± 18.7	70.0 ± 20.2	48.5 ± 32.2
	TTN	-73.9 ± 4.3	-68.7 ± 10.1	-61.6 ± 15.0	-56.0 ± 17.9	-52.3 ± 20.0	-30.7 ± 32.9
	Hybrid-TTN	-7.2 ± 2.9	-7.9 ± 6.4	-6.4 ± 8.5	-3.2 ± 11.9	+0.4 ± 15.8	+5.4 ± 21.9
3 classes	Source	93.4 ± 1.6	87.2 ± 11.0	80.0 ± 13.3	71.8 ± 15.6	69.0 ± 17.5	52.3 ± 23.0
	TTN	-28.0 ± 3.0	-26.7 ± 8.1	-21.2 ± 9.6	-16.3 ± 14.0	-14.4 ± 14.4	+0.2 ± 20.9
	Hybrid-TTN	-2.2 ± 0.9	-1.0 ± 3.5	+1.6 ± 5.4	+5.1 ± 9.0	+9.1 ± 12.4	+13.0 ± 15.1
5 classes	Source	94.2 ± 1.1	87.4 ± 8.5	81.2 ± 10.7	75.4 ± 13.8	65.4 ± 19.5	51.9 ± 23.2
	TTN	-15.3 ± 1.6	-10.8 ± 5.2	-6.3 ± 6.7	-2.3 ± 10.2	+3.7 ± 15.1	+14.6 ± 21.5
	Hybrid-TTN	-1.7 ± 0.5	+0.3 ± 2.9	+3.2 ± 5.2	+6.8 ± 7.8	+10.6 ± 10.7	+16.4 ± 16.1

Table 1: CIFAR-10 evaluations on multiple label shifted distributions and covariate shifts (corruptions) with different degrees of label imbalance. We show the source model accuracy and the improvement (or degradation) as a delta accuracy. We observe that the proposed method provides benefits over source model when there is no covariate shift, while avoiding catastrophic failures and allowing benefits over source when there are label distribution shifts.

		Covariate Shift					
		Accuracy (% or Δ%)					
Label Distribution Shift		Original	Corruption-1	Corruption-2	Corruption-3	Corruption-4	Corruption-5
Original (1000 classes)	Source	69.5 ± 2.0	52.5 ± 7.7	42.2 ± 10.6	33.1 ± 14.2	23.0 ± 14.8	14.8 ± 12.6
	TTN	-0.3 ± 0.8	+6.6 ± 3.6	+8.4 ± 5.0	+10.3 ± 6.6	+11.5 ± 6.7	+11.0 ± 7.1
	Hybrid-TTN	-0.2 ± 0.8	+2.6 ± 3.1	+3.2 ± 4.3	+4.1 ± 5.6	+4.3 ± 4.1	+4.2 ± 4.1
1 class	Source	71.8 ± 18.9	52.6 ± 19.6	45.1 ± 26.2	29.3 ± 21.1	18.6 ± 21.6	15.1 ± 18.9
	TTN	-70.3 ± 19.1	-51.3 ± 19.7	-43.9 ± 26.2	-28.0 ± 21.2	-17.6 ± 21.5	-14.3 ± 19.1
	Hybrid-TTN	-11.6 ± 10.8	-10.7 ± 14.0	-10.2 ± 14.8	-9.3 ± 16.3	-5.9 ± 15.1	-4.5 ± 14.0
5 classes	Source	67.3 ± 8.5	51.7 ± 11.6	43.3 ± 14.2	32.4 ± 15.3	23.8 ± 15.0	15.4 ± 13.5
	TTN	-28.0 ± 4.9	-20.2 ± 6.7	-16.2 ± 9.8	-10.9 ± 10.5	-6.1 ± 9.4	-1.4 ± 9.6
	Hybrid-TTN	-1.9 ± 2.0	+0.3 ± 5.2	+0.6 ± 5.8	+1.7 ± 7.9	+1.2 ± 6.8	+2.3 ± 5.5

Table 2: ImageNet-1K evaluations on multiple label shifted distributions and covariate shifts (corruptions) with different degrees of label imbalance. We observe that the proposed method provides benefits over source model when there is covariate shift, while avoiding catastrophic failures when there are label distribution shifts.

4 Conclusions

We have studied a popular batch-level Test-time Adaptation method in the context of label distribution shift. We observed that in realistic scenarios where batches at deployment time have label distribution shifts, this method can fail catastrophically. We proposed a direction for solving this problem to keep the benefits of adaptation without risking catastrophic failure due to label shift.

Acknowledgments and Disclosure of Funding

This work was supported by grants from the Institute of Data Valorization (IVADO PRF3) to A.T., G.W. and E.B. A.T. acknowledges support from the Fonds de Recherche du Québec–Santé (FRQ-S) and the Fondation de l'Association des Radiologistes du Québec (FARQ) Clinical Research Scholarship–Senior Salary Award (FRQS-ARQ no. 298509). G.W. acknowledges support from the Canada CIFAR AI Chair. E.B. and G.W. acknowledge funding from Fonds de recherche du Québec — Nature et technologies - NOVA (2023-NOVA-329759 and 2023-NOVA-329125).

References

[1] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[2] Jin Kim, Jiyoung Lee, Jungin Park, Dongbo Min, and Kwanghoon Sohn. Pin the memory: Learning to generalize semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4350–4360, June 2022.

[3] Vibashan VS, Poojan Oza, and Vishal M Patel. Towards online domain adaptive object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 478–488, 2023.

[4] Michael Blaivas, Laura N Blaivas, and James W Tsung. Deep learning pitfall: Impact of novel ultrasound equipment introduction on algorithm performance and the realities of domain adaptation. *Journal of Ultrasound in Medicine*, 41(4):855–863, 2022.

[5] Yan Wang, Yangqin Feng, Lei Zhang, Zizhou Wang, Qing Lv, and Zhang Yi. Deep adversarial domain adaptation for breast cancer screening from mammograms. *Medical Image Analysis*, 73:102147, 2021.

[6] 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 Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9229–9248. PMLR, 13–18 Jul 2020.

[7] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *arXiv preprint arXiv:2303.15361*, 2023.

[8] 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*, pages 295–305, 2022.

[9] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

[10] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[11] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[12] Zachary Nado, Shreyas Padhy, D. Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *CoRR*, abs/2006.10963, 2020.

[13] Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. TTN: A domain-shift aware batch normalization in test-time adaptation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[16] Collin Burns and Jacob Steinhardt. Limitations of post-hoc feature alignment for robustness. *CoRR*, abs/2103.05898, 2021.

[17] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[18] Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to imagenet. In *International conference on machine learning*, pages 583–593. PMLR, 2019.

[19] Edouard Oyallon. Building a regular decision boundary with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5106–5114, 2017.

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

[21] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition*, 2009.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, June 2016.

[24] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016.

[25] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

A Appendix

A.1 Hybrid-TTN algorithm

Algorithm block for the proposed Hybrid Test Time Normalization.

Algorithm 1: Hybrid-TTN

Input: Trained network f with K layers; source model statistics $\{\mu_k^s, \sigma_k^s\}_{k=1..K}$; data batch x_0.

for k *in* $\{1, ..., K\}$ **do**
  $x_k = f_{k-1}(x_{k-1})$
  $\mu_k^t, \sigma_k^t \leftarrow$ COMPUTE BN STATS FROM x_k
  $R = \frac{k}{K}$
  for c *in* C *channels* **do**
    SCORES[c] $= W_2^2([\mu_i^s]_c, [\sigma_i^s]_c, [\mu_i^t]_c, [\sigma_i^t]_c)$
  end
  TOPR-IND $\leftarrow$ COMPUTE TOP-R INDEX(SCORES)
  $m_k[\text{TOPR-IND}] = 0$
  $m_k[\text{TOP}\tilde{\text{R}}\text{-IND}] = 1$
  $\mu_k^{hybrid} = m_k \odot \mu_k^t + (1 - m_k) \odot \mu_k^s$
  $\sigma_k^{hybrid} = m_k \odot \sigma_k^t + (1 - m_k) \odot \sigma_k^s$
  $x_k = B(x_k, \mu_k^{hybrid}, \sigma_k^{hybrid})$
end
Output: $\{\mu_k^{hybrid}, \sigma_k^{hybrid}\}_{k=1..K}$

A.2 Ablation

In order to validate our proposed method, we perform an ablation study aimed at investigating the efficacy of channel selection strategy within Hybrid-TTN. Specifically, we explore an alternative approach where the $T\%$ percentage of channels to be adapted per layer are randomly selected, as opposed to using the sorted distances to determine a threshold (see Section 2).

The results, shown in Table A1, indicate that the random selection of channels does not yield good outcomes, as the model is severely affected by the distribution shift. This ablation validates our premise that the selection of channels with the lowest Wasserstein distance is an effective strategy.

	Covariate Shift					
	Accuracy (% or Δ%)					
	Original	Corruption-1	Corruption-2	Corruption-3	Corruption-4	Corruption-5
Source	94.5 ± 3.3	87.2 ± 10.9	80.8 ± 14.7	74.7 ± 18.7	70.0 ± 20.2	48.5 ± 32.2
Hybrid-TTN	-7.2 ± 2.9	-7.9 ± 6.4	-6.4 ± 8.5	-3.2 ± 11.9	+0.4 ± 15.8	+5.4 ± 21.9
Random Channel-TTN	-21.9 ± 6.9	-25.7 ± 11.6	-22.6 ± 14.3	-18.4 ± 18.3	-13.2 ± 22.6	-5.2 ± 28.1

Table A1: CIFAR-10 ablation. Using random channels instead of the sorted channels in the Hybrid-TTN. It is notable that selecting random channels is detrimental to the performance of the adapted models, as one would intuitively expect.