Robustifying Vision Transformer without Retraining from Scratch by Test-Time Class-Conditional Feature Alignment

Takeshi Kojima* , **Yutaka Matsuo** and **Yusuke Iwasawa**
The University of Tokyo, Japan
{t.kojima,matsuo,iwasawa}@weblab.t.u-tokyo.ac.jp

Abstract

Vision Transformer (ViT) is becoming more popular in image processing. Specifically, we investigate the effectiveness of test-time adaptation (TTA) on ViT, a technique that has emerged to correct its prediction during test-time by itself. First, we benchmark various test-time adaptation approaches on ViT-B16 and ViT-L16. It is shown that the TTA is effective on ViT and the prior-convention (sensibly selecting modulation parameters) is not necessary when using proper loss function. Based on the observation, we propose a new test-time adaptation method called class-conditional feature alignment (CFA), which minimizes both the class-conditional distribution differences and the whole distribution differences of the hidden representation between the source and target in an online manner. Experiments of image classification tasks on common corruption (CIFAR-10-C, CIFAR-100-C, and ImageNet-C) and domain adaptation (digits datasets and ImageNet-Sketch) show that CFA stably outperforms the existing baselines on various datasets. We also verify that CFA is model agnostic by experimenting on ResNet, MLP-Mixer, and several ViT variants (ViT-AugReg, DeiT, and BeiT). Using BeiT backbone, CFA achieves 19.8% top-1 error rate on ImageNet-C, outperforming the existing test-time adaptation baseline 44.0%. This is a state-of-the-art result among TTA methods that do not need to alter training phase.[1]

1 Introduction

Inspired by the success in natural language processing, Transformer [Vaswani *et al.*, 2017] is becoming more and more popular in various image processing tasks, including image recognition [Dosovitskiy *et al.*, 2020; Touvron *et al.*, 2021], object detection [Carion *et al.*, 2020], and video processing [Zhou *et al.*, 2018; Zeng *et al.*, 2020]. Notably, [Dosovitskiy *et al.*, 2020] proposed Vision Transformer (ViT), which adapts Transformer architecture to image classification tasks and shows that it achieves comparable or superior performance to that of the conventional convolutional neural networks (CNNs). Follow-up research also shows that ViT is more robust to the common corruptions and perturbations than convolution-based models (e.g., ResNet) [Paul and Chen, 2021; Morrison *et al.*, 2021], which is an important property for safety-critical applications.

This study seeks to answer the following question: *can we improve the robustness of ViT without retraining it from scratch*? Most prior works focused on how to robustify the models during training. For example, [Hendrycks *et al.*, 2019; Hendrycks *et al.*, 2020] demonstrated that several data augmentation improves robustness of convolutional neural networks (CNN). Similarly, [Chen *et al.*, 2022] shows that a sharpness-aware optimizer improves the robustness of ViT. Unfortunately, such approaches require retraining the models from scratch, which entails a massive computational burden and training time for large models (such as ViT). Moreover, sometimes dataset for pre-training is not publicly available, which makes it impossible to retrain the models.

This study investigates the effectiveness of test-time adaptation (TTA) to robustify ViT. TTA is a recently emerged approach for improving the robustness of models without retraining them from scratch and accessing to training dataset [Schneider *et al.*, 2020; Nado *et al.*, 2020; Wang *et al.*, 2020]. Instead, it corrects the model's prediction for test data by modulating its parameters during test time. For example, [Wang *et al.*, 2020] proposed Tent, which modulates the parameters of batch normalization (BN) by minimizing prediction entropy. It was shown that Tent can significantly improve the robustness of ResNet. TTA has two major advantages over usual training-time techniques. First, it does not alter the training phase and thus does not need to repeat the computationally heavy training phase. Second, it does not require accessing to the source data during adaptation, which is impossible in the case of large pre-trained models.

Conceptually speaking, TTA can be applied to arbitrary network architectures. However, naively modulating model parameters during test-time may cause a catastrophic failure as discussed in [Wang *et al.*, 2020]. To avoid the issue, prior works often limited the modulation parameters, which resulted in architecture constraints. For example, [Schneider *et al.*, 2020; Wang *et al.*, 2020] modulated statistics and/or

*Contact Author

[1]Full version (with Appendix) and code are publicly available at https://github.com/kojima-takeshi188/CFA.

affine transformation in batch normalization (BN) layer, but the BN-based method cannot be applied to some modern models such as ViT since they do not have BN.

This study contributes to addressing this research question by following two means. First, we benchmark various test-time adaptation methods on ViT using several robustness benchmark tasks (CIFAR10-C, CIFAR100-C, ImageNet-C, and several domain adaptation tasks). We also design modulation parameters potentially suitable for ViT-based architectures. The tested methods include entropy minimization [Wang *et al.*, 2020], pseudo-classifier [Iwasawa and Matsuo, 2021], pseudo-label [Lee and others, 2013], diversity regularization [Liang *et al.*, 2020], and feature alignments [Liu *et al.*, 2021]. Regarding modulation parameters, we sweep over the following four candidates: layer normalization [Ba *et al.*, 2016], CLS token, feature extractor [Liang *et al.*, 2020], and entire parameters of a model. The results indicate that the prior convention in test-time adaptation (i.e., limiting modulation parameters) *is not necessary* when using proper loss function, while it is necessary for pure entropy minimization based approach. This observation is important for applying TTA to arbitrary network architectures.

We then propose a new loss function: test-time class-conditional feature alignment (CFA). Our approach can be categorized into feature alignment approach as with [Liu *et al.*, 2021], which minimize the gap of the statistics between training and test domain. It is worth noting that the feature alignment approach for test-time adaptation (e.g., our approach and [Liu *et al.*, 2021]) assumes that one can access to the statistics on the source dataset during the test phase but does not need to access to the source dataset itself and to repeat the computationally heavy training [2]. Therefore, this approach can be used without source dataset during adaptation. We show that such complementary information about the source data distribution can stabilize the training without selecting the modulation parameters. In addition, we extend the feature alignment approach [Liu *et al.*, 2021] by the following two means. First, CFA aligns class-conditional statistics as well as the statistics of overall distribution. Second, we calculate the statistics after properly normalizing the hidden representations. Despite the simplicity, these techniques significantly boost the performance of test-time adaptation.

In summary, our main contributions are as follows.

- This is the first study that verifies the effectiveness of test-time adaptation methods on ViT. By benchmarking several test-time adaptation approaches under common corruptions and domain adaptation tasks, we have validated that the robustness of ViT model is improved during test time without retraining the model from scratch.
- We introduce a new test-time adaptation method (CFA). Throughout the experiment, CFA achieves better results than existing baselines on multiple datasets. In addition, CFA is robust to hyperparameter tuning, which is important in practically setting up test-time adaptation.
- We show that CFA consistently improves the robustness on a wide variety of backbone networks during test time. In particular, we achieve the state-of-the-art results of test-time adaptation on ImageNet-C with a 19.8 % top-1 error rate when using BeiT-L16 as a backbone network.

[2] Test-time adaptation generally assumes that the model would be distributed without source data due to bandwidth, privacy, or profit reasons [Wang *et al.*, 2020]. We argue that the statistics of source data would be distributed even in such a situation since it could drastically compress data size and eliminate sensitive information. In fact, some layers often used in typical neural networks contain statistics of source data (e.g., batch normalization)

2 Related Work

2.1 Vision Transformer (ViT)

Transformer [Vaswani *et al.*, 2017], first proposed in natural language processing (NLP) field, also achieves great performance in image processing as Vision Transformer (ViT) [Dosovitskiy *et al.*, 2020]. ViT divides input image data into small patches and translates them to embedding vectors, otherwise known as a "token". Extra learnable class embedding (CLS token) is added to the sequence of the tokens before feeding them into Transformer Encoder. Transformer Encoder mainly consists of multilayered global self-attention blocks and MLP blocks. The blocks include layer normalization [Ba *et al.*, 2016] as one function. An MLP head is added to top layer of the CLS token as a classifier.

Since its invention, ViT has rapidly become popular in the field of computer vision. Many applications and extensions have been proposed thus far. For example, [Touvron *et al.*, 2021],[Bao *et al.*, 2021], and [Steiner *et al.*, 2021] showed that the performance of ViT is respectively improved by distillation (DeiT), self-supervised learning (BeiT), and data augumentation (ViT-AugReg) during pre-training phase. More recently, [Tolstikhin *et al.*, 2021] proposed MLP-Mixer, which was proven to be quite competitive with ViT by replacing self-attention blocks with MLP layers.

This study is interested in how to robustify ViT to the common perturbations. Recent experimental research has verified that ViT inherently has robustness without any adaptation or any additional data augmentation. Several studies empirically show that ViT is inherently more robust than CNNs [Paul and Chen, 2021; Morrison *et al.*, 2021; Naseer *et al.*, 2021] by using some benchmark datasets. Several studies have shown that the robustness of ViT can be improved by changing the training strategy, such as using a larger data set for the pre-training phase [Paul and Chen, 2021; Bhojanapalli *et al.*, 2021] or a sharpness-aware optimizer for the training phase [Chen *et al.*, 2022]. However, retraining such a massively pre-trained model from scratch is not desirable considering the computational burden. The larger the data and model, the higher it costs for retraining. At the same time, however, the model size also matters for robustness, i.e., larger models tend to be more robust by themselves (See Appendix E for the detail). This observation motivated us to investigate a lightweight and model-agnostic way to improve the robustness of the models.

2.2 Test-Time Adaptation (TTA)

This study investigates the effectiveness of the test-time adaptation approaches for Vision Transformer and its variants. Unlike most existing works that focus on training phase to

improve the robustness, test-time adaptation focuses on test-time. In other words, test-time adaptation does not alter the training phase; therefore, we do not need to repetitively run computationally heavy training to improve robustness.

The algorithm of existing test-time adaptation can be summarized by following two aspects: (1) adaptation function f_{adapt} and (2) modulation parameters $\boldsymbol{\psi}$. Literally, f_{adapt} is a function that determines how to modulate the model parameter during the test time. More formally, f_{adapt} receives a batch of unlabeled images $\boldsymbol{X}_{test}$, which is available online at test-time, and updates the target parameter $\boldsymbol{\psi}$ using the data. A naive instance of f_{adapt} might use stochastic gradient decent (SGD) by designing a loss function that can effectively incorporate $\boldsymbol{X}_{test}$ to correct its prediction. For example, Tent [Wang *et al.*, 2020], which is a pioneering method for test-time adaptation, minimizes prediction entropy using SGD, based on the assumption that a more confident prediction (i.e. low prediction entropy) leads to a more accurate prediction. One can also use different loss functions (such as pseudo-label (PL) [Lee and others, 2013], diversity regularization (SHOT-IM) [Liang *et al.*, 2020], and feature alignments (TFA) [Liu *et al.*, 2021]), or design optimization-free procedures to update the model (e.g., T-BN [Schneider *et al.*, 2020; Nado *et al.*, 2020] and T3A [Iwasawa and Matsuo, 2021]).

The second aspect is the selection of modulation parameters $\boldsymbol{\psi}$. As discussed in [Wang *et al.*, 2020], updating the entire model parameters $\boldsymbol{\theta}$ is often ineffective in test-time optimization because $\boldsymbol{\theta}$ is usually the only information of the source data in the setup, and updating all parameters without restriction results in catastrophic failure. (See Table 1 for the experiment result). Consequently, prior works also proposed to sensibly select modulation parameters along with the adaptation method f_{adapt}. For example, [Schneider *et al.*, 2020; Nado *et al.*, 2020] proposed to re-estimate the statistics of batch normalization [Ioffe and Szegedy, 2015] during the test time while fixing the other parameters. Similarly, Tent [Wang *et al.*, 2020] modulated only a set of affine transformation parameters of the BN layer. This causes two problems when applying test-time adaptation to ViT. First, ViT has significantly larger parameters compared to ResNet which is the standard test bed of prior studies. Consequently, the effectiveness of TTA on such a huge model has not been fully investigated. Second, ViT and its variants do not have BN, so they cannot directly take advantage of the common good strategy. In other words, there is a lack of knowledge regarding which parameter should be updated to effectively robustify ViT.

In this study, we avoid the difficulty of sensibly selecting the modulation parameters by incorporating the feature-alignment approach. More specifically, we explicitly minimize the difference between some statistics of source distribution and target (test) distribution, rather than simply modulating model parameters only given data from target distribution. In other words, we leverage the source statistics as auxiliary information regarding the source distribution to prevent adaptation from causing the aforementioned catastrophic failure. Note that our method does not rely on the co-existence of source and target data and does not violate the setting of test-time adaptation.

Similar to our work, [Liu *et al.*, 2021] recently proposed test-time feature alignment (TFA), which aligns the hidden representation between source and target data by minimizing the distance of the mean and covariance matrix. Our method is different from TFA in the following two aspects. First, we propose to align class-conditional statistics as well as the statistics of overall distribution. Second, we propose to calculate the statistics after properly normalizing the hidden representations. In §4.4, our experiment results demonstrate that these techniques stably improve the performance of various tasks based on various backbone networks.

3 Methods

3.1 Modulation Parameters

As discussed in §2.2, the choice of modulation parameters is regarded as important in test-time adaptation but prior BN-based modulation is not applicable to Vision Transformer. To find good candidates for $\boldsymbol{\psi}$ in ViT, we sweep over the following four candidates: layer normalization [Ba *et al.*, 2016], CLS token, feature extractor parameters, and all parameters.

A Layer normalization (LN) re-estimates the mean and standard deviation of input across the dimensions of the input, followed by the affine transformation for each dimension. We update the affine transformation parameters in LN for adaptation. A CLS token is a parameterized vector and proven to be efficient for fine-tuning large models for downstream tasks in NLP [Lester *et al.*, 2021]. A feature extractor is defined as any module in a model except for its classifier. This term is borrowed from [Liang *et al.*, 2020]. In the case of ViT, its feature extractor consists of Transformer Encoder, patch embeddings, and positional embeddings. Updating feature extractor parameters is a basic unsupervised domain adaptation setting, while [Wang *et al.*, 2020] claimed that it was ineffective in test-time adaptation setup (see §2.2).

It is worth noting that these choices of modulation parameters are applicable to many modern architectures, including ViT, DeiT, MLP-Mixer, and BeiT. This property is important in practice because a better backbone network usually provides significant performance gains. We also empirically show the effectiveness of TTA on such various architectures.

3.2 Class-Conditional Feature Alignment

Regarding the adaptation function, this study proposes a new loss function, called class-conditional feature alignment (CFA). Similar to the most prior works, our method uses stochastic gradient decent to adapt the model during test-time. Unlike the prior methods such as Tent, PL, and T3A that modulate the parameters using the data available at test-time only, our method aligns the statistics of features between source and target. In other words, we leverage the source statistics as an auxiliary information regarding the source distribution to prevent the model from suffering a catastrophic failure.

Assume that a model consists of two components; a linear classifier g_ω as last layer, and a feature extractor f_ϕ before the classifier. A set of source training samples is denoted as $X^s = \{x_i^s\}_{i=1}^{N_s}$. While prior works often calculate the statistics of feature output by f_ϕ, the feature is not always normalized. For example, ViT uses GELU as an activation function

Algorithm 1 Online Adaptation using CFA

Input: Fine-tuned DNN model with parameters θ, Partial parameters to be updated during adaptation $\psi \subset \theta$, Target test dataset X^t, m-th ordered batch data $X^{t,m} \subset X^t$, Statistics of Eq.(2) (3) (4) calculated from source training dataset.
Output:

1: **for** $m = 1$ to M **do**
2: Predict labels $\hat{Y}^{t,m}$ for $X^{t,m}$
3: Calculate statistics Eq.(6) (7) (8) for $X^{t,m}$
4: Update ψ using Eq.(11)
5: **end for**
6: **return** $(\hat{Y}^{t,1}, .., \hat{Y}^{t,M})$

and LN with elementwise affine transformation before classifier, which is not bounded. We found that this causes unstable behavior especially when matching higher order moments of distributions. Thus, before calculating the statistics, we normalize (bound the minimum and maximum value of) the hidden representation for each sample $f_\phi(x_i^s)$ as follows.

$$h(x_i^s) = Tanh\left(LN^\dagger\left(f_\phi(x_i^s)\right)\right), \tag{1}$$

where $LN^\dagger$ is defined as layer normalization *without affine transformation*. Despite the simplicity, we empirically find that not only matching higher order moment of overall distribution is stabilized, but also the performance of class-conditional feature alignment is boosted (See Table 5 for the detail). The feature normalization might have a positive effect on class-conditional distribution matching by highlighting the distribution property of each class.

After the normalization, the mean and higher order central moments of overall distribution on source data are calculated and stored in memory as fixed values.

$$\mu^s = \frac{1}{|X^s|} \sum_{x_i^s \in X^s} h(x_i^s), \tag{2}$$

$$\mathbb{M}_k^s = \frac{1}{|X^s|} \sum_{x_i^s \in X^s} (h(x_i^s) - \mu^s)^k, \quad (k = 2, ..., K) \tag{3}$$

where K denotes the maximum number of moments. Class-conditional mean of the normalized hidden representations is also calculated and stored in memory as fixed value as follows

$$\mu_c^s = \frac{1}{|X_c^s|} \sum_{x_i^s \in X_c^s} h(x_i^s), \qquad (c = 1, ..., C) \tag{4}$$

where C denotes the number of classes. $X_c^s \subset X^s$ contains all the source samples whose ground-truth labels are c. Note that these statistics are calculated before adaptation, i.e., we do not need to access to source data itself in test phase.

CFA uses these statistics to adapt the model during test phase. Assume that a sequence of test data drawn from target distribution arrives at our model one after another. Test dataset is denoted as $X^t = \{x_i^t\}_{i=1}^{N_t}$, and a set of test data in m-th batch is denoted as $X^{t,m} \subset X^t, (m = 1, ..., M)$. For each batch, hidden representations of test data are normalized and their statistics are calculated in the same way as source.

$$h(x_i^t) = Tanh\left(LN^\dagger\left(f_\phi(x_i^t)\right)\right), \tag{5}$$

ViT-B16	Tent	PL	SHOT-IM	CFA
LN	**50.6±0.5**	**55.7±1.4**	45.7±0.0	43.9±0.0
CLS	59.4±0.0	60.6±0.0	59.9±0.0	58.2±0.0
Feature	56.2±2.2	60.8±2.1	**43.9±0.0**	**41.8±0.0**
ALL	59.1±1.0	61.4±2.2	44.0±0.0	**41.8±0.0**
ViT-L16	Tent	PL	SHOT-IM	CFA
LN	**42.3±0.0**	**44.3±0.0**	42.0±0.0	40.2±0.0
CLS	50.3±0.0	51.3±0.0	50.7±0.1	49.2±0.0
Feature	43.8±0.6	46.5±0.8	**38.4±0.0**	**36.6±0.0**
ALL	44.2±1.1	46.9±0.7	**38.4±0.0**	**36.6±0.0**

Table 1: Modulation parameter choice study. The evaluation metric is top-1 error on ImageNet-C averaged over 15 corruption types with severity level of 5. ViT-B16 and ViT-L16 are used as the models. CLS: CLS token, LN: layernorm params, Feature: parameters of feature extractor, ALL: all the parameters of ViT.

$$\mu^{t,m} = \frac{1}{|X^{t,m}|} \sum_{x_i^t \in X^{t,m}} h(x_i^t), \tag{6}$$

$$\mathbb{M}_k^{t,m} = \frac{1}{|X^{t,m}|} \sum_{x_i^t \in X^{t,m}} (h(x_i^t) - \mu^{t,m})^k, \ (k = 2...K) \tag{7}$$

$$\mu_c^{t,m} = \frac{1}{|X_c^{t,m}|} \sum_{x_i^t \in X_c^{t,m}} h(x_i^t), \qquad (c = 1, ..., C) \tag{8}$$

where $X_c^{t,m}$, which is a subset of $X^{t,m}$, includes all samples in the current batch annotated as class c by pseudo-labeling $argmax_c\, g_\omega(f_\phi(x_i^t))$. In this study, the overall distribution distance is defined by the central moment distance (CMD) [Zellinger *et al.*, 2017] (see Appendix G for details):

$$\mathcal{L}_F = \frac{1}{2}||\mu^s - \mu^{t,m}||_2 + \frac{1}{2^k} \sum_{k=2}^{K} ||\mathbb{M}_k^s - \mathbb{M}_k^{t,m}||_2. \tag{9}$$

As for class-conditional distribution matching, following prior studies in UDA setting ([Xie *et al.*, 2018; Deng *et al.*, 2019]), we use class-conditional centroid alignment.

$$\mathcal{L}_C = \frac{1}{2|C'|} \sum_{c \in C'} ||\mu_c^s - \mu_c^{t,m}||_2, \tag{10}$$

where C' denotes a set of the pseudo labelled classes belonging to the current target minibatch samples. The first-order moment (centroid) is sufficient for class-conditional feature alignment when class size is larger than batch size. Parameters ψ of the model are updated by the gradient of the following loss function based on the target batch data at hand.

$$\mathcal{L} = \mathcal{L}_F + \lambda \mathcal{L}_C, \tag{11}$$

where λ is a balancing hyperparameter. Following [Wang *et al.*, 2020], for efficient computation, we use the scheme that the parameter update follows the prediction for the current batch. Therefore, the update only affects the next batch. The adaptation procedure is summarized in Algorithm 1.

Method	Type	Class Cond.	C10→ C10-C	C100→ C100-C	ImageNet→ ImageNet-C	SVHN→ MNIST	SVHN→ MNIST-M	ImageNet→ ImageNet-S
Source	-		14.6±0.0	35.1±0.0	61.9±0.0	23.2±0.0	46.2±0.0	64.1±0.0
T3A	gf		13.7±0.0	34.0±0.0	61.2±0.0	17.4±0.3	40.9±0.2	61.7±0.0
Tent	fm		10.9±0.2	27.4±0.5	50.6±0.5	15.3±0.2	53.0±1.7	68.3±4.3
PL	fm		11.9±0.0	30.1±0.5	55.7±1.4	15.8±0.7	49.7±1.9	62.2±1.2
SHOT-IM	fm		8.9±0.0	25.6±0.0	45.7±0.0	**13.7±0.1**	36.6±0.4	**56.1±0.1**
TFA(-)	fa		8.8±0.0	32.2±0.2	57.8±0.1	16.5±0.1	39.3±0.4	65.7±0.2
CFA-F (Ours)	fa		**8.7±0.0**	**25.2±0.0**	46.7±0.0	16.3±0.0	39.9±0.1	57.2±0.1
CFA-C (Ours)	fa	✓	**8.5±0.0**	**25.3±0.1**	**45.3±0.0**	14.2±0.0	**35.8±0.2**	57.6±0.1
CFA (Ours)	fa	✓	**8.4±0.0**	**24.6±0.1**	**43.9±0.0**	14.2±0.1	**36.3±0.2**	**56.1±0.0**

Table 2: Method comparison on each adaptation tasks. The evaluation metric is top-1 error rate. The results of CIFAR-10-C, CIFAR-100-C and ImageNet-C are ones averaged over the 15 corruption types with highest severity level (=5). CFA-F : Overall distribution matching only. CFA-C : Class-conditional distribution matching only. gf : Gradient free method. fm : Method that controls the output feature representation (without depending on feature alignment) by modulation. fa : Method that utilizes feature alignment between source and target by modulation. Our proposal (CFA-C and CFA) is the only method utilizing the class-conditional feature alignment during test-time adaptation.

4 Experiment

4.1 Datasets and Task Design

Common Corruptions. We validate the robustness against common corruptions on CIFAR-10-C, CIFAR-100-C and ImageNet-C [Hendrycks and Dietterich, 2019] as target datasets. These datasets contain data with 15 types corruptions with five levels of severity, that is, each dataset has 75 distinct corruptions. Most of our experiments use the highest severity(=5) datasets as they can make the difference in performance most noticeable. As source datasets, CIFAR-10, CIFAR-100 [Krizhevsky and Hinton, 2009] and ImageNet(-2012) [Russakovsky *et al.*, 2015] are used, respectively.

Domain Adaptation. We validate the robustness against style shift on small-sized datasets and large-sized datasets. For small-sized datasets, we evaluate the adaptation from SVHN to MNIST / MNIST-M [Netzer *et al.*, 2011; LeCun *et al.*, 1998; Ganin and Lempitsky, 2015]. For large-sized datasets, we evaluate the adaptation from ImageNet to ImageNet-Sketch [Wang *et al.*, 2019]. See Appendix A for a detail description of each dataset.

4.2 Implementation Details

Vistion Transformer (ViT-B16) is used as a default model throughout the experiment unless an explicit explanation is provided. Images of all the datasets are resized to 224×224 (see Appendix B for details). Before adaptation, the model is fine-tuned on each source dataset (see Appendix C for details). In addition, the central moments statistics of hidden representation based on source data need to be calculated to store them in memory. For this purpose, we use all the training data in the source dataset and set the dropout [Srivastava *et al.*, 2014] off in the model during the calculation.

As for default hyperparameters for adaptation on target data, batch size is set as 64, optimizer is set as SGD with a constant learning rate of 0.001, and momentum of 0.9 with gradient clipping [Zhang *et al.*, 2020] at global norm 1.0 across all the experiments (Gradient clipping has the effect of preventing adaptation by Tent from catastrophic failure in the severe corruption setting. See "Ablation Study" in §4.4). As for CFA, the balancing parameter λ is set as 1.0, and maximum central moments order K is set as 3. During prediction and parameter update, dropout is set off in models.

As an evaluation metric, top-1 error of classification is used across all the experiments. We run all the experiments three times with different seeds for different data ordering by shuffling. A mean and unbiased standard deviation of the metric are reported. Our implementation is in PyTorch [Paszke *et al.*, 2019]. We use various backbone networks from timm library [Wightman, 2019] and torchvision library (Appendix D). Every experiment is run on cloud A100 x 1GPU instance.

4.3 Baseline Methods

We compare CFA with some existing baseline test-time adaptation methods that do not need to alter training phase as described in §2.2: **Tent, PL, TFA(-)**[3] , **T3A and SHOT-IM**. In addition, we report the performance of the model on target datasets without any adaptation as **Source**. T-BN is excluded from the baseline because some models (ViT variants and MLP-Mixer) do not have a batch normalization layer. For a fair comparison, we use the same hyperparameters across all the methods as described in §4.2.

4.4 Experiment Result

Modulation Study. Table 1 answers the question about which set of modulation parameters is the most suitable for improving the performance of test-time adaptations on ViTs. There are two findings. First, updating layer nomalization parameters can achieve balanced and high performance across all the main methods. Second, SHOT-IM and CFA achieve

[3]Original TFA needs to alter training phase (add contrastive learning), while this study focuses on robustifying large-scale models without retraining them from scratch. Therefore, we have changed some of the settings from the original TFA so that the model does not need to alter training phase. The modified version of TFA is denoted as TFA(-) in our experiments. See Appendix F for details.

	ImageNet-C	ImageNet-S
ResNet50	82.0±0.0	75.4±0.0
+ CFA / SHOT-IM	**58.8±0.0**/**58.8±0.0**	70.0±0.2/**69.2±0.1**
ResNet101	77.4±0.0	72.3±0.0
+ CFA / SHOT-IM	**55.3±0.1**/55.7±0.0	66.8±0.0/**66.2±0.1**
ViT-B16	61.9±0.0	64.1±0.0
+ CFA / SHOT-IM	**43.9±0.0**/45.7±0.0	**56.0±0.1**/56.1±0.1
ViT-L16	53.4±0.0	59.1±0.0
+ CFA / SHOT-IM	**40.2±0.0**/42.0±0.0	**52.6±0.0**/53.6±0.1
DeiT-S16	59.9±0.0	66.6±0.0
+ CFA / SHOT-IM	**46.0±0.0**/46.1±0.0	60.3±0.1/**59.4±0.0**
DeiT-B16	52.9±0.0	62.5±0.0
+ CFA / SHOT-IM	**39.9±0.0**/**39.9±0.0**	55.9±0.0/**55.4±0.0**
MLP-Mixer-B16	73.3±0.0	74.3±0.0
+ CFA / SHOT-IM	**52.4±0.1**/55.1±0.1	**64.2±0.1**/65.9±0.2
MLP-Mixer-L16	77.1±0.0	79.8±0.0
+ CFA / SHOT-IM	**56.3±0.0**/62.4±0.1	**70.8±0.3**/72.9±0.3
ViT-B16-AugReg	49.0±0.0	57.0±0.0
+ CFA / SHOT-IM	**37.6±0.0**/38.4±0.0	51.5±0.1/**51.0±0.2**
ViT-L16-AugReg	39.1±0.0	48.2±0.0
+ CFA / SHOT-IM	**32.1±0.0**/33.3±0.0	**45.2±0.0**/45.6±0.1
BeiT-B16	48.3±0.0	52.6±0.0
+ CFA / SHOT-IM	**35.4±0.0**/37.6±0.0	**47.5±0.0**/49.1±0.0
BeiT-L16	35.9±0.0	44.2±0.0
+ CFA / SHOT-IM	**26.0±0.0**/28.2±0.0	**39.9±0.1**/41.5±0.0

Table 3: Adaptation results based on several backbone networks. The evaluation metric of ImageNet-C is the averaged top-1 error over 15 corruption types with a severity level of 5. We use publicly available models that were already fine-tuned on ImageNet.

higher performance by updating all or feature extractor parameters, while Tent and PL deteriorates the performance because of catastrophic failure (See Appendix I for details). This indicates that a method with more sophisticated strategy within the adaptation function can work properly without sensibly selecting modulation parameters. In all the subsequent experiments, we choose layer normalization as modulation parameters across all the methods for the fair comparison.

CFA Outperforms Existing Methods on Several Datasets. Table 2 summarizes the adaptation result across datasets for each test time adaptation methods. As for CIFAR-10-C, CIFAR-100-C, and ImageNet-C, we measure the averaged top-1 error across 15 corruption types for the highest severity level (=5). CFA (our method) aligns both the overall distribution and class-conditional distribution between source and target datasets. In addition to CFA, we have experimented class-conditional distribution matching only method (CFA-C) and overall distribution matching only method (CFA-F) to measure the contribution of each distribution matching to performance. Specifically, the objective function of CFA-C and CFA-F is respectively defined as Eq.(10) and Eq.(9). The experiment results demonstrate that CFA can achieve the best or comparable performance against baseline methods across all datasets. It is also verified that CFA-F and CFA-C can solely achieve better performance compared to the case without adaptation ("Source") as in Table 2. Finally, CFA further boosts the performance on most datasets by combining them.

Severity	Source	Tent	SHOT-IM	CFA
1	16.8±0.0	15.4±0.0	15.8±0.0	**15.3±0.0**
2	20.3±0.0	17.9±0.0	18.4±0.0	**17.5±0.0**
3	22.5±0.0	19.6±0.4	19.9±0.0	**18.7±0.0**
4	27.2±0.0	24.0±1.2	23.2±0.0	**21.5±0.0**
5	35.9±0.0	33.6±0.1	28.2±0.0	**26.0±0.0**
Average	24.5±0.0	22.1±0.3	21.1±0.0	**19.8±0.0**

Table 4: Top-1 error rate on ImageNet-C averaged across all the severity level and 15 corruption types. BeiT-L16 is used as a model.

Method	W/. Eq(1)(5)	W/O. Eq(1)(5)
Source	61.95±0.00	61.95±0.00
CFA-F (K=1)	46.69±0.02	46.69±0.01
CFA-F (K=3)	46.66±0.02	47.28±0.03
CFA-F (K=5)	46.64±0.02	54.51±0.14
CFA-C	**45.31±0.03**	47.13±0.06
CFA (K=1)	43.98±0.04	45.28±0.02
CFA (K=3)	**43.90±0.04**	44.56±0.03
CFA (K=5)	**43.90±0.04**	52.25±0.11

Table 5: Ablation study of CFA. Top-1 error on ImageNet-C averaged over 15 corruption types with severity level of 5. ViT-B16 is used. CFA-F : Overall distribution matching only. CFA-C : Class-conditional distribution matching only. K : Maximum # of central moments. K=1 denotes first-order moment (mean) matching only.

CFA Is Model Agnostic. Table 3 shows the adaptation results on ImageNet-C and ImageNet-Sketch by CFA (and SHOT-IM for comparison) based on various category's backbone networks. Specifically, we used publicly available models that are already fine-tuned on ImageNet-2012 at a resolution of 224 $\times$ 224, including ResNet, ViT, ViT-AugReg, DeiT, BeiT, and MLP-Mixer. See Appendix D for details. The modulation parameters are BN for ResNet, and LN for the others. The results indicate that our method (CFA) consistently improves the performance regardless of the backbone networks. It is also found that the better performance on the source dataset (ImageNet), the stronger robustness on the target dataset (ImageNet-C) the model can gain by adaptation. See Appendix E for visualization of the relationship.

CFA Achieves SOTA Performance. Among these backbone networks, we select BeiT-L16, which achieved strong performance on ImageNet, and calculate the top-1 error rate on ImageNet-C averaged over 15 types of corruptions and all the severity levels (1-5) for each TTA methods. The results described in Table 4 demonstrate that 19.8% using CFA on BeiT-L16 gives superior performance to the other baseline methods. It also outperforms the existing test-time adaptation result 44.0% using Tent on ResNet50 [Wang *et al.*, 2020]. Therefore, CFA achieves the state-of-the-art (SOTA) performance among TTA methods that do not need to alter training phase (See Appendix I for the full results).

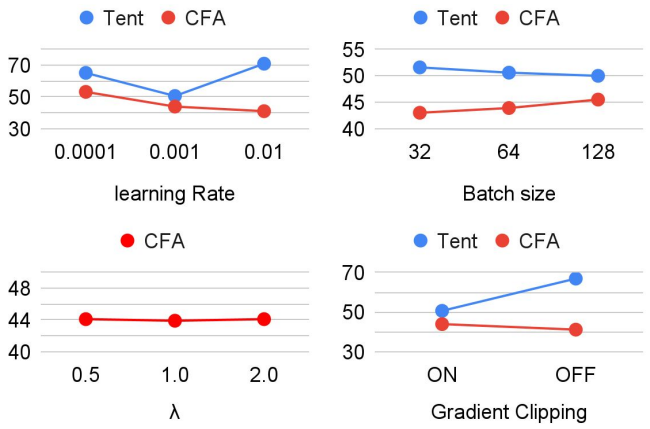

Figure 1: The effect of changing hyperparameters on Tent and CFA performance. The evaluation metric is the top-1 error on ImageNet-C averaged over 15 corruption types with a severity level of 5. ViT-B16 is used as a model. Either one of the hyperparameter values is changed from the default described in §4.2.

Ablation Study. Table 5 summarizes the ablation study results to analyze the detailed contributions of each components in our method on the robustness. Specifically, we analyze the effect of the normalization of hidden representation before calculating the distribution statistics by comparing the scenarios of with/without Eq.(1) and (5). In the case of CFA-F, it is verified that the performance deteriorates significantly without Eq.(1) and (5) especially when the maximum number of moments K gets larger. This indicates that feature normalization, especially bounding minimum and maximum value of hidden representation, stabilizes the performance of matching higher order moments. In the case of CFA-C, it is verified that the performance deteriorates without Eq.(1) and (5). It is speculated that feature normalization, especially layer normalization without affine transformation, might have a positive effect on class-conditional (centroid) distribution matching by highlighting the distribution property for each class. In addition, it is also verified that using both overall feature alignment and class-conditional feature alignment (CFA) boosts the performance compared to either alone (CFA-F or CFA-C) regardless of the value of K.

Hyperparameter Sensitivity. For online adapatation, hyperparameter selection is a challenging issue. Figure 1 shows the experiment results about each hyperparameter sensitivity on ImageNet-C with the highest severity level (=5) averaged over 15 corruption types. We checked 4 hyperparameters by changing either one of the values from the default described in §4.2. (a) learning rate, (b) batch size, (c) balancing hyperparameter λ, and (d) whether to enable gradient clipping for SGD optimization. The finding is that Tent is more sensitive to some hyperparameters than CFA. In particular, enabling gradient clipping is essential when applying Tent to ViT to avoid catastrophic failure, while it is not essential for CFA. Furthermore, large learning rate also causes Tent catastrophic failure. In contrast, CFA is robust to all the above hyperparameters. This indicates that we can safely use CFA in unknown environments with rough hyperparameter selection.

5 Conclusion

This is the first study that verifies the effectiveness of test-time adaptation methods on ViT to boost their robustness. Experiment results demonstrate that the existing methods can be applied to ViT and the prior-convention (sensibly selecting modulation parameters) is not necessary when a proper loss function is used. This study also proposed a novel method, CFA, which is hyperparameter friendly, model agnostic, and surpasses existing baselines. We hope this study becomes a milestone of TTA for current large models and will serve as a stepping stone to TTA for larger models in the future.

Acknowledgements

This work has been supported by the Mohammed bin Salman Center for Future Science and Technology for Saudi-Japan Vision 2030 at The University of Tokyo (MbSC2030). Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used for experiments.

References

[Ba *et al.*, 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[Bao *et al.*, 2021] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[Bhojanapalli *et al.*, 2021] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF ICCV*, pages 10231–10241, 2021.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020.

[Chen *et al.*, 2022] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. In *ICLR*, 2022.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[Deng *et al.*, 2019] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *Proceedings of ICCV*, pages 9944–9953, 2019.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[Ganin and Lempitsky, 2015] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189. PMLR, 2015.

[Hendrycks and Dietterich, 2019] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proc. of the ICLR*, 2019.

[Hendrycks *et al.*, 2019] 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.

[Hendrycks *et al.*, 2020] 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. *arXiv preprint arXiv:2006.16241*, 2020.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456. PMLR, 2015.

[Iwasawa and Matsuo, 2021] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *Advances in NeurIPS*, 2021.

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

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Lee and others, 2013] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[Lester *et al.*, 2021] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of EMNLP*, pages 3045–3059, 2021.

[Liang *et al.*, 2020] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, pages 6028–6039. PMLR, 2020.

[Liu *et al.*, 2021] Yuejiang Liu, Parth Kothari, Bastien Germain van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. TTT++: When does self-supervised test-time training fail or thrive? In *Advances in NeurIPS*, 2021.

[Morrison *et al.*, 2021] Katelyn Morrison, Benjamin Gilby, Colton Lipchak, Adam Mattioli, and Adriana Kovashka. Exploring corruption robustness: Inductive biases in vision transformers and mlp-mixers. *arXiv preprint arXiv:2106.13122*, 2021.

[Nado *et al.*, 2020] 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.

[Naseer *et al.*, 2021] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *arXiv preprint arXiv:2105.10497*, 2021.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop*, 2011.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in NeurIPS*, pages 8026–8037, 2019.

[Paul and Chen, 2021] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. *arXiv preprint arXiv:2105.07581*, 2(3), 2021.

[Rusak *et al.*, 2020] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *ECCV*, pages 53–69. Springer, 2020.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[Schneider *et al.*, 2020] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in NeurIPS*, 33, 2020.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The JMLR*, 15(1):1929–1958, 2014.

[Steiner *et al.*, 2021] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

[Tolstikhin *et al.*, 2021] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[Touvron *et al.*, 2021] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in NeurIPS*, pages 5998–6008, 2017.

[Wang *et al.*, 2019] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in NeurIPS*, pages 10506–10518, 2019.

[Wang *et al.*, 2020] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2020.

[Wightman, 2019] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. Accessed: 2021-12-27.

[Xie *et al.*, 2018] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *ICML*, pages 5423–5432, 2018.

[Zellinger *et al.*, 2017] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for domain-invariant representation learning. In *5th ICLR*, 2017.

[Zeng *et al.*, 2020] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *ECCV*, pages 528–543. Springer, 2020.

[Zhang *et al.*, 2020] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *ICLR*, 2020.

[Zhou *et al.*, 2018] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on CVPR*, pages 8739–8748, 2018.