CAFA: Class-Aware Feature Alignment for Test-Time Adaptation

Sanghun Jung[1] Jungsoo Lee[2] Nanhee Kim[3] Amirreza Shaban[1] Byron Boots[1] Jaegul Choo[2]
[1] University of Washington, [2]KAIST AI, [3]Enssel Inc.

Abstract

Despite recent advancements in deep learning, deep neural networks continue to suffer from performance degradation when applied to new data that differs from training data. Test-time adaptation (TTA) aims to address this challenge by adapting a model to unlabeled data at test time. TTA can be applied to pretrained networks without modifying their training procedures, enabling them to utilize a well-formed source distribution for adaptation. One possible approach is to align the representation space of test samples to the source distribution (i.e., feature alignment). However, performing feature alignment in TTA is especially challenging in that access to labeled source data is restricted during adaptation. That is, a model does not have a chance to learn test data in a class-discriminative manner, which was feasible in other adaptation tasks (e.g., unsupervised domain adaptation) via supervised losses on the source data. Based on this observation, we propose a simple yet effective feature alignment loss, termed as Class-Aware Feature Alignment (CAFA), which simultaneously 1) encourages a model to learn target representations in a class-discriminative manner and 2) effectively mitigates the distribution shifts at test time. Our method does not require any hyper-parameters or additional losses, which are required in previous approaches. We conduct extensive experiments on 6 different datasets and show our proposed method consistently outperforms existing baselines.

1. Introduction

Recent advancements [17, 51, 11, 10] in machine learning are effective in solving diverse problems, achieving remarkable performance enhancements on benchmark datasets. However, these methods can suffer from significant performance degradation when applied to test data with different properties from the training data (*i.e.,* source data), such as corruption [18], changing lighting conditions [8], or adverse weather [53, 6]. Sensitivity to distribution shifts [41] hampers deep networks from performing well in practical scenarios where test samples may differ from training data [26]. Thus, adapting deep models to the test samples is crucial when distribution shifts exist.

Various adaptation methods [3, 15, 36, 14, 48, 13, 55, 21] have been proposed to alleviate this problem. However, most of these methods require either access to the source data during adaptation [48, 14, 19] or modification of the training procedure [33, 32, 49], which limits their applicability. Therefore, we seek to design an adaptation method that 1) is applicable to existing deep networks without modification and 2) does not require access to the source data during adaptation. Satisfying such conditions, previous studies perform adaptation at test time while making predictions simultaneously, which is referred to as *test-time adaptation* (TTA).

One widely adopted approach to address distribution shifts is to align the source (*i.e.,* training data) and target (*i.e.,* test data) distributions [14, 48, 34, 47, 13, 50]. For example, DANN [14] directly reduces the $\mathcal{H}$-divergence between the source and target distributions, and CORAL [48] minimizes the difference in the second-order statistics between the source and target data. Despite their demonstrated effectiveness in the unsupervised domain adaptation (UDA) task, applying those alignments to TTA has the following limitation. Alignments are generally performed along with supervised losses on the source data, which encourages a model to learn target distributions in a class-discriminative manner [48]. However, access to the source data is prohibited during adaptation in TTA, precluding learning class discriminability.

With this issue in mind, we conduct an analysis of the effects of feature alignments in TTA by using two distances in the representation space: intra-class distance and inter-class distance. As shown in Fig. 1 (c), intra-class distance (dotted arrow) is defined as the distance between a sample and its ground-truth source class distribution, and inter-class distance (solid arrow) denotes the averaged distance between the sample and the other source class distributions. Achieving low intra-class distance and high inter-class distance is crucial for improving classification accuracy [30, 4, 31, 42, 56].

For analysis, we first adopt a feature alignment that reduces the domain-level discrepancy between source and target domains, which is a commonly adopted paradigm in UDA studies [14, 48, 47]. One straightforward approach to achieve this is to align the mean and covariance of the

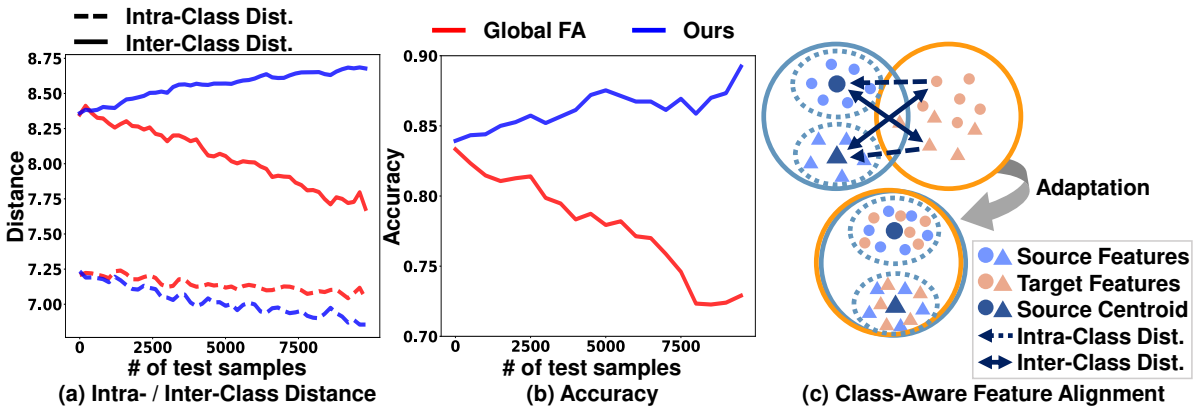

Figure 1. A motivating example of our paper. (a) shows the change of the intra-class distance (dotted lines) and the inter-class distance (solid lines) of ours (blue line) and global feature alignment (red line). (b) shows the accuracy changes as adaptation proceeds. (c) illustrates how our method CAFA aligns the test features to the source distribution in a class-discriminative manner. We obtain the plots by adapting a model to corrupted images of the CIFAR10-C dataset. Please refer to Section 4 for further details.

source and target distributions, *i.e.,* global feature alignment (Global FA).[1] The result of applying this feature alignment in TTA is depicted in Fig. 1 (a) (red line). The intra-class distance is reduced, which is desirable but is also accompanied by a decrease in inter-class distance. Such effects can degrade the image classification accuracy in Fig. 1 (b) (red line). This is mainly due to the lack of class information in the global feature alignment. A model does not have a chance to learn the test data in a class-discriminative manner since a supervised loss is not available on both the source and target data.

Motivated by such observations, we propose Class-Aware Feature Alignment (CAFA) that aligns the target features to the pre-calculated source feature distributions by considering both intra- and inter-class distances. To be more specific, we pre-calculate the statistics (*i.e.,* mean and covariance) of the source distribution to estimate class-conditional Gaussian distributions from a pretrained network. At test time, we use the Mahalanobis distance [37] to 1) align each sample to its predicted class-conditional Gaussian distribution (*i.e.,* reduce intra-class distance) and 2) enforce samples to be distinct from the other class-conditional Gaussian distributions (*i.e.,* increase inter-class distance). Applying CAFA successfully enhances class discriminability as shown in Fig. 1 (a) (blue line) and significantly improves the classification accuracy as adaptation proceeds (Fig. 1 (b) (blue line)). We empirically show that reducing intra-class distance alone is not sufficient as it could also reduce the inter-class distance and result in performance degradation.

Aligning feature distributions at test time requires access to the source data *before adaptation* to pre-calculate the source statistics, as similarly done in previous methods [12, 33, 7]. However, we empirically show that CAFA only requires a small number of training samples (*e.g.,* 5% of the training samples in the ImageNet/CIFAR10 datasets (Fig. 3)) to obtain robust source statistics that outperform the existing methods. In addition, CAFA does not require any hyper-parameters or modifications on pretraining procedures for adaptation.

The main contributions of our work are as follows:

- We propose a novel Class-Aware Feature Alignment (CAFA) that effectively mitigates distribution shifts and encourages a model to learn discriminative target representations simultaneously.
- Our proposed approach is simple yet effective, not requiring hyper-parameters or additional modifications of the training procedure.
- We conduct extensive experiments on 6 different datasets along with in-depth analyses and show that CAFA consistently outperforms the existing methods on test-time adaptation.

2. Related Work

2.1. Test-time Adaptation

Existing UDA approaches [2, 1, 3, 15, 36, 40, 43, 19] have addressed distribution shifts effectively by adapting to target domains at training time. UDA approaches generally assume that 1) source data is available during adaptation, and 2) we already know which target domain the models are adapted to. However, these assumptions sometimes do not hold in real-world scenarios. To address such concerns, approaches that adapt a model at test time have been proposed, not requiring access to the source data during adaptation [54, 20, 57, 58, 44, 29, 49, 33, 23]. Several methods [49, 33] perform adaptation in an offline manner, predicting test samples after iterating multiple epochs over the entire set of the test samples (*i.e.,* test-time training). These approaches modify the training procedure to have self-supervised losses (*e.g.,* rotation prediction or contrastive

[1] We will explain the global feature alignment in more detail in Section 3.2

loss) and utilize them as proxy losses for adaptation. However, as also pointed out in Wang *et al.* [54], it is not guaranteed that optimizing the proxy losses helps in improving the main task since they are not directly related to classifying images into categories. Addressing such concerns, test-time adaptation (TTA) methods [54, 20, 57, 58, 44, 5] have been proposed. These approaches do not require any modification of the training procedures, allowing the algorithms to be applicable to a given pretrained deep learning network. TENT [54], a recent seminal work in TTA, proposed to update the modulation parameters in batch normalization [22] layers while minimizing the entropy loss, effectively mitigating distribution shifts.

2.2. Feature Alignment

Feature alignment is widely adopted in UDA studies to mitigate distribution shifts [47, 13, 34, 50]. However, most of these approaches do not consider categorical information but rather match the source and target distributions globally. This may harm class discrimination performance since it does not guarantee class-to-class matching between two distributions [4]. Tacking the problem, various studies have proposed to align distributions in a class-discriminative manner [4, 31, 35, 16, 42, 59, 46]. This point of view is also relevant to test-time adaptation, and we design an effective loss that simultaneously mitigates the distribution gap while improving class discriminability.

3. Proposed Method

3.1. Preliminary

Assume that we have a model $f_s(x) = h_s \circ g_s(x)$ pretrained with a supervised loss $\mathcal{L}(x_s, y_s)$ on source data, where $x_s \in \mathcal{X}_s$ and $y_s \in \mathcal{Y}_s$. Here $g_s : \mathcal{X}_s \rightarrow \mathbb{R}^d$ denotes the pretrained feature extractor, and $h_s : \mathbb{R}^d \rightarrow \mathbb{R}^C$ indicates the pretrained classifier, where d is the dimension of extracted features, and C is the number of classes. Then, we aim to adapt the pretrained model $f_s(\cdot)$ to target data x_t while correctly classifying them at test time.

Mahalanobis distance In this work, we adopt the Mahalanobis distance [37] to align the source and target distributions. The Mahalanobis distance measures the distance between a distribution and a sample. With an input image x, feature extractor $g(\cdot)$, and Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the Mahalanobis distance is defined as

$$D(x; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (g(x) - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (g(x) - \boldsymbol{\mu}). \quad (1)$$

Intra-/inter-class distance For analysis, we measure the intra- and inter-class distances between the class-conditional source distributions and target samples. To be more specific, we define class-conditional Gaussian distributions as $P(g_s(x)|y = c) = \mathcal{N}(g_s(x)|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, where $\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c$ are the mean and covariance of the multivariate Gaussian distribution of class $c \in \{1, ..., C\}$. Then, with a target image x_t, the intra-class distance is defined as

$$D_{\text{intra}}(x_t, y_t) = D(x_t; \boldsymbol{\mu}_{y_t}, \boldsymbol{\Sigma}_{y_t}), \quad (2)$$

where y_t indicates the corresponding ground-truth label of the target image. Analogously, the inter-class distance is defined as

$$D_{\text{inter}}(x_t, y_t) = \frac{1}{C-1} \sum_{c=1}^{C} \mathbb{1}(y_t \neq c) D(x_t; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \quad (3)$$

Note that achieving low intra-class distance and high inter-class distance is important for improving image classification accuracy.

3.2. Analysis of Class Discriminability in Feature Alignment

We compare and analyze three different feature alignments with respect to the intra- and inter-class distances. First, we investigate the global feature alignment (Global FA) that reduces the discrepancy between the source and target distributions without considering class information. With the given source Gaussian distribution $\mathcal{N}(g_s(x_s)|\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$, the Global FA loss $\mathcal{L}_{\text{FA}}$ is formulated as

$$\mathcal{L}_{\text{FA}} = \|\boldsymbol{\mu}_s - \hat{\boldsymbol{\mu}}_t\|_2^2 + \|\boldsymbol{\Sigma}_s - \hat{\boldsymbol{\Sigma}}_t\|_F^2, \quad (4)$$

where $\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s$ denote the mean and covariance of source features without considering class information, and $\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\Sigma}}_t$ indicate the mean and covariance estimated from a mini-batch of test samples. $\|\cdot\|_2, \|\cdot\|_F$ denote the Euclidean norm and Frobenius norm, respectively. As shown in Fig. 1 (a) (red lines), while the Global FA drastically decreases the intra-class distance, it accompanies a significant reduction of inter-class distance which needs to be high for achieving a reasonable level of image classification accuracy. Fig. 1 (b) (red line) also verifies such a point by visualizing the degraded image classification accuracy.

To address such a problem, we take the *class information* into account when aligning features. Aligning the source and target distributions in a class-wise manner would be one straightforward approach. However, at test time, there exist very few samples for each class in a mini-batch to precisely estimate class-conditional distributions of test data. Thus, we align individual test samples to the source class-conditional distribution of the predicted classes by using the Mahalanobis distance. Note that we adopt the predicted class of each sample as a proxy of its ground truth label since we do not have access to the true label [29]. Specifically, with the given source class-conditional Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, the loss $\mathcal{L}_{\text{intra}}$ minimizing the intra-class distance is defined as

$$\mathcal{L}_{\text{intra}} = \frac{1}{N} \sum_{n=1}^{N} D_{\text{intra}}(x_{t,n}, \hat{y}_{t,n}), \quad (5)$$

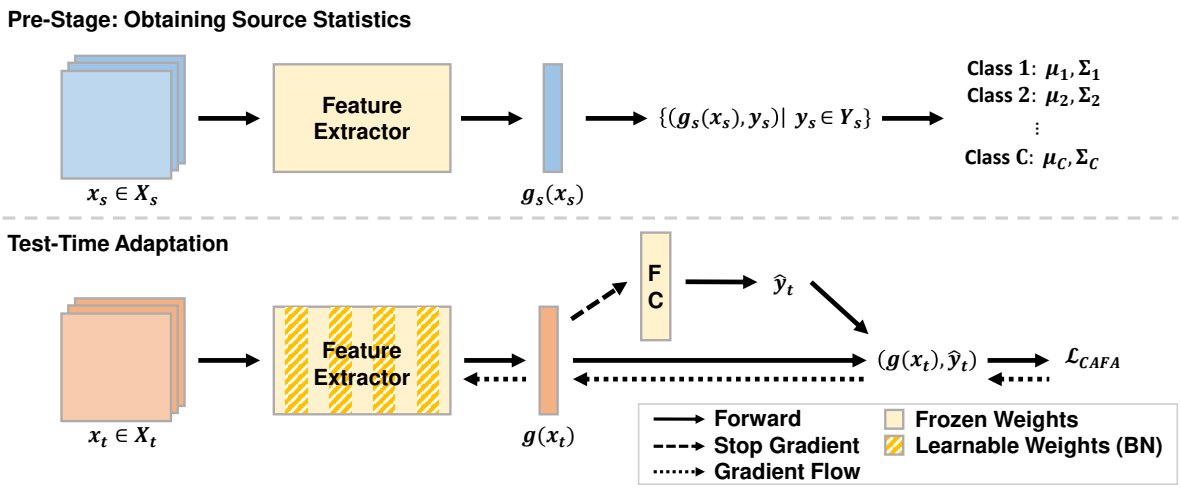

Figure 2. Overview of our method. (Pre-stage) Our method first pre-calculates source class-conditional Gaussian distributions using a pre-trained network. (Test-time adaptation) During test-time adaptation, we adapt a model by optimizing class-aware feature alignment loss while making predictions simultaneously.

where $\hat{y}_{t,n}$ indicates the predicted class of the target sample $x_{t,n}$, and N denotes the number of target samples. While utilizing $\mathcal{L}_{\text{intra}}$ effectively reduces the intra-class distance, it still decreases the inter-class distance as shown in Fig. 4 (a) (green line). We present our method in the next section that addresses this issue by adopting the loss function to also enlarge the inter-class distance.

3.3. CAFA: Class-Aware Feature Alignment

Pre-calculation of source statistics As shown in the pre-stage of Fig. 2, we calculate C class-conditional Gaussian distributions $P(g_s(x_s)|y=c) = \mathcal{N}(g_s(x_s)|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ with the pretrained feature extractor $g_s(\cdot)$ over source training samples (x_s, y_s) with the following equations:

$$\begin{aligned} \boldsymbol{\mu}_c &= \frac{1}{N_c}\sum_{n=1}^{N_c} g_s(x_{s,n_c}), \\ \boldsymbol{\Sigma}_c &= \frac{1}{N_c}\sum_{n=1}^{N_c}(g_s(x_{s,n_c}) - \boldsymbol{\mu}_c)(g_s(x_{s,n_c}) - \boldsymbol{\mu}_c)^\top, \end{aligned} \tag{6}$$

where N_c denotes the number of training samples of class c, and x_{s,n_c} indicates training samples of class c.

Test-time adaptation With the source class-conditional distributions $P(g_s(x_s)|y=c)$, we perform class-aware feature alignment at test time as illustrated in the test-time adaptation stage of Fig. 2. We initialize a model using the weights of pretrained networks $f_s(\cdot)$ and perform adaptation to target data considering both intra- and inter-class distances. Our final loss $\mathcal{L}_{\text{CAFA}}$ is defined as

$$\mathcal{L}_{\text{CAFA}} = \frac{1}{N}\sum_{n=1}^{N} \log \frac{D_{\text{intra}}(x_{t,n}, \hat{y}_{t,n})}{\sum_{c=1}^{C} D(x_{t,n}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}. \tag{7}$$

As shown in Fig. 4 (a) (blue line), our final loss aligns the source and target distributions in a desirable way by reducing the intra-class distance and enlarging the inter-class distance.

3.4. Theoretical Background

Gaussian assumption of features Recent studies [37, 28] present theoretical justifications about the Gaussian assumption of features when the network is trained with the Softmax function. For image classification, a discriminative classifier is trained using the Softmax function whose posterior distribution is

$$p(y=c|x) = \frac{\exp(w_c^\intercal x + b_c)}{\sum_{c'}\exp(w_{c'}^\intercal x + b_{c'})}, \tag{8}$$

where x, y denote input features and labels, and w_c, b_c indicate weight and bias. However, a generative classifier such as Gaussian discriminant analysis (GDA) can also be used for classification. GDA defines posterior distribution by assuming that a class distribution follows the multivariate Gaussian distribution $p(x|y=c) = \mathcal{N}(x|\mu_c, \Sigma_c)$, and a class prior distribution follows the Bernoulli distribution $p(y=c) = \frac{\beta_c}{\sum_{c'}\beta_{c'}}$. Additionally, GDA assumes all the class-conditional distributions share the same covariance, *i.e.*, $\Sigma_c = \Sigma$. The posterior distribution of GDA is represented as

$$\begin{aligned} p(y=c|x) &= \frac{p(y=c)p(x|y=c)}{\sum_{c'} p(y=c')p(x|y=c')} \\ &= \frac{\exp(\mu_c^\intercal\Sigma^{-1}x - \frac{1}{2}\mu_c^\intercal\Sigma^{-1}\mu_c + \log\beta_c)}{\sum_{c'}\exp(\mu_{c'}^\intercal\Sigma^{-1}x - \frac{1}{2}\mu_{c'}^\intercal\Sigma^{-1}\mu_{c'} + \log\beta_{c'})}. \end{aligned} \tag{9}$$

The posterior distribution of GDA becomes equivalent to the one from the Softmax function if we set the weight

Method	Gaus.	Shot	Impu.	Defo.	Glas.	Moti.	Zoom	Snow	Fros.	Fog	Brig.	Cont.	Elas.	Pixe.	Jpeg.	Average
Source	48.73	44.00	57.00	11.84	50.78	23.38	10.84	21.93	28.24	29.41	7.01	13.27	23.38	47.88	19.46	29.14
BN	17.34	16.36	28.25	9.89	26.11	14.27	8.15	16.29	13.82	20.69	8.58	8.49	19.67	11.74	14.17	15.59
PL	17.22	16.07	27.85	9.74	25.94	14.13	8.07	16.12	13.78	20.14	8.53	8.53	19.73	11.65	13.94	15.43
FR-Online†	17.23	16.15	27.31	10.07	25.58	14.12	8.35	16.17	13.67	20.01	8.64	8.65	19.48	11.82	14.20	15.43
TFA-Online†	15.80	14.91	23.89	9.29	23.08	12.82	7.41	13.93	12.60	16.41	7.43	7.95	17.24	12.00	12.86	13.84
TTT++-Online†	16.80	14.92	21.99	9.60	22.97	12.32	7.55	13.14	12.67	14.33	7.06	7.85	17.27	11.63	12.74	13.52
TENT†	15.95	14.55	24.72	9.03	23.25	12.74	7.47	13.91	12.78	16.66	8.13	8.12	18.30	10.85	13.21	13.98
EATA†	16.73	15.42	25.09	9.83	24.10	13.36	8.45	15.02	13.64	17.39	8.63	8.44	19.08	11.70	13.97	14.72
CAFA (Ours)	**14.28**	**12.70**	**21.12**	**7.73**	**20.84**	**10.55**	**6.75**	**11.93**	**11.31**	**13.33**	**6.95**	**7.13**	**16.08**	**9.59**	**11.67**	**12.13**
Source	80.77	77.84	87.75	39.62	82.26	54.22	38.38	54.58	60.19	68.11	28.86	50.93	59.54	72.27	49.96	60.35
BN	47.37	45.58	60.10	34.01	56.70	40.99	32.05	46.53	42.57	54.41	32.56	33.30	48.83	37.47	39.43	43.46
PL	46.74	45.26	59.21	33.83	56.08	40.29	31.64	46.10	42.07	53.74	32.24	33.08	48.24	37.11	39.01	43.00
FR-Online†	47.16	45.60	59.85	34.09	56.70	41.06	32.20	46.44	42.65	54.37	32.72	33.48	48.85	37.49	39.45	43.47
TFA-Online†	44.68	43.28	56.17	32.47	54.11	37.48	30.32	42.46	39.73	47.57	30.18	32.52	45.34	36.81	37.28	40.69
TTT++-Online†	43.70	41.84	55.77	31.15	53.38	35.54	29.98	41.13	38.70	45.08	29.14	30.34	44.69	35.47	37.37	39.55
TENT†	43.11	41.70	53.30	31.35	51.08	36.34	29.90	42.73	38.99	45.13	29.64	30.62	44.03	34.23	36.34	39.23
EATA†	43.12	41.94	52.20	32.02	50.35	36.56	30.42	41.94	39.31	43.52	29.88	30.89	44.75	34.55	37.10	39.24
CAFA (Ours)	**41.60**	**39.77**	**50.45**	**30.17**	**48.35**	**34.65**	**28.76**	**39.52**	**37.42**	**41.25**	**27.95**	**29.54**	**42.37**	**32.87**	**35.02**	**37.31**

Table 1. Classification error (%) on the CIFAR10-C (upper group) and CIFAR100-C (lower group) datasets with severity level 5 corruptions. † denotes the results obtained from the official codes.

Method	Averaged Error (%) ↓
ResNet-26 (GroupNorm)	32.70
• MEMO [58]	29.68
ResNet-26 (GroupNorm)+JT	35.30
• TTT [49]	20.00
• TTT (Episodic)	32.85
ResNet-26 (BatchNorm)	34.93
• FR-Online [12]	18.43
• TENT [54]	17.25
• CAFA (Ours)	**16.72**

Table 2. Classification error (%) on the CIFAR10-C dataset with severity level 5 corruptions using ResNet-26 networks. All the numbers are obtained from the official codes. JT denotes joint training.

$w_c = \mu_c^\intercal \Sigma^{-1}$ and the bias $b_c = -\frac{1}{2}\mu_c^\intercal \Sigma^{-1}\mu_c + \log \beta_c$. This derivation implies that source features x may follow the Gaussian distribution when the network is trained with the Softmax function.

Interpretation of CAFA Even though our loss is devised from intuition (Fig. 1), we can interpret our loss by using the posterior distribution of GDA. By assuming a uniform prior distribution and assuming each class has its own covariance, the negative log-posterior can be simplified as

$$
\begin{aligned}
&-\log p(y=c|x) \qquad (10)\\
&= -\log \frac{\exp(-\frac{1}{2}D(x;\mu_c,\Sigma_c) - \frac{1}{2}\log|\Sigma_c|)}{\sum_{c'}\exp(-\frac{1}{2}D(x;\mu_{c'},\Sigma_{c'}) - \frac{1}{2}\log|\Sigma_{c'}|)}.
\end{aligned}
$$

While this term and our loss both play a similar role in minimizing/maximizing the intra-/inter-class distances, we observed that the above term generates large gradients since it can get easily saturated due to the high variance of Mahalanobis distances. On the other hand, we empirically found that our final loss is more stable, and thus, allows us to use a higher learning rate, achieving state-of-the-art performance. From these analyses, we believe our loss generalizes well if the source model is trained with the Softmax function.

4. Experiments

This section first demonstrates evaluation results in two different settings: 1) robustness to corruptions, and 2) domain adaptation beyond image corruptions. Then, we present in-depth analyses of our method by conducting ablation studies and providing visualizations of the representation space.

Baselines For evaluations, we consider the following baselines in our experiments: no adaptation (Source), test-time normalization (BN) [44], pseudo label (PL) [29], test-time training (TTT) [49], test-time entropy minimization (TENT) [54], efficient anti-forgetting test-time adaptation (EATA) [38], constrastive test-time-adaptation (AdaContrast) [5], test-time template adjuster (T3A) [23], marginal entropy minimization with one test-point (MEMO) [58], feature restoration (FR-Online) [12], test-time feature alignment (TFA-Online) [33], and test-time training++ (TTT++-Online) [33][2]. For fair comparisons, we note that FR-Online, TFA-Online, TTT++-Online, and CAFA store the training statistics before deployment for test-time adaptation while others do not require such a step. Further details about the baselines can be found in the supplementary materials.

Implementation details We adopt the ResNet50 [17] for our main experiments except for Table 2. For the test-time adaptation, we set the batch size as 200 and utilize the Adam [24] optimizer with a learning rate of 0.001 for adaptation. For ImageNet-C experiments, we adopt a batch size of 64, and we set a learning rate to 0.0025 for CAFA

[2]Online adaptation denotes predicting incoming test samples immediately, while offline adaptation predicts the test samples after several iterations of the entire test data.

Method	Gaus.	Shot	Impu.	Defo.	Glas.	Moti.	Zoom	Snow	Fros.	Fog	Brig.	Cont.	Elas.	Pixe.	Jpeg.	Average
Source	94.86	93.14	97.25	86.96	88.45	77.87	77.63	83.71	80.35	92.85	81.94	98.55	69.03	58.89	55.64	82.47
BN	68.74	67.80	74.72	68.71	77.13	61.04	59.71	67.37	67.21	76.52	66.09	93.73	61.32	55.07	55.75	68.06
PL	68.11	66.95	74.18	67.92	76.52	60.32	58.87	67.08	66.63	75.99	65.34	93.38	60.82	54.77	55.50	67.49
TENT†	64.11	63.72	70.35	63.22	73.39	56.64	55.07	64.28	62.99	70.39	60.88	**92.44**	57.17	51.72	52.94	63.95
EATA†	65.18	64.20	72.84	63.71	74.81	56.90	55.81	65.28	64.43	71.79	60.11	97.54	57.98	52.20	53.85	65.11
CAFA (Ours)	**63.68**	**63.04**	**70.12**	**61.27**	**71.30**	**55.22**	**54.34**	**63.31**	**61.88**	**67.96**	**59.15**	92.53	**56.16**	**51.21**	**52.60**	**62.92**

Table 3. Classification error (%) on the TinyImageNet-C dataset with severity level 5. † denotes the results obtained from the official codes.

Method	Gaus.	Shot	Impu.	Defo.	Glas.	Moti.	Zoom	Snow	Fros.	Fog	Brig.	Cont.	Elas.	Pixe.	Jpeg.	Average
Source	97.79	97.07	98.15	82.08	90.18	85.22	77.50	83.11	76.69	75.57	41.07	94.57	83.05	79.39	68.35	81.99
BN	84.73	84.11	84.16	84.95	84.83	73.61	61.12	65.66	66.90	51.86	34.77	83.18	55.90	51.17	60.43	68.49
PL	85.66	88.98	84.34	93.01	91.91	65.85	54.46	55.59	69.07	43.48	32.75	98.85	47.67	41.88	48.61	66.81
TENT†	71.26	69.54	69.99	71.95	72.85	58.77	50.69	52.86	58.89	42.57	32.68	73.29	45.17	41.57	47.94	57.33
EATA†	**65.00**	**63.10**	**64.30**	**66.30**	**66.60**	**52.90**	**47.20**	**48.60**	**54.30**	**40.10**	**32.00**	**55.70**	**42.40**	**39.30**	**45.00**	**52.00**
CAFA (Ours)	69.59	67.29	68.03	71.09	70.87	56.13	50.03	50.77	56.77	41.86	33.24	61.30	43.76	40.87	47.03	55.24

Table 4. Classification error (%) on the ImageNet-C dataset with severity level 5. † denotes the results obtained from the official codes.

and 0.00025 for others with SGD optimizer. Note that we only optimize the modulation parameters γ, β in batch normalization layers, following Wang *et al.* [54].

4.1. Robustness to Corruptions

Datasets For corruption datasets, we evaluate methods on the CIFAR10-C, CIFAR100-C, TinyImageNet-C, and ImageNet-C [18] datasets. CIFAR10 [25] and CIFAR100 [25] include 50,000 training samples and 10,000 test samples with 10 and 100 classes, respectively. TinyImageNet [27] is a subset of the original ImageNet [9] dataset, containing 100,000 training images and 10,000 validation images with 200 classes. ImageNet [9] contains 1.2 million training samples and 50,000 validation samples with 1,000 object categories. CIFAR10-C and CIFAR100-C [18] datasets contain 15 different corruptions, and the corruptions are applied to the test set of CIFAR10 and CIFAR100 datasets. Analogous to the CIFAR10-C / CIFAR100-C datasets, TinyImageNet-C and ImageNet-C datasets are composed of 15 corruption types, where the corruptions are applied to the validation set of TinyImageNet and ImageNet, respectively.

Quantitative evaluation and comparisons To evaluate the robustness to corruption, we utilize the pretrained networks on CIFAR10, CIFAR100, TinyImageNet, and ImageNet datasets and adapt the pretrained networks to their corresponding corruption datasets, respectively.

Table 1 shows the image classification errors (%) on diverse corruption types with the severest corruption level in the CIFAR10-C and CIFAR100-C datasets. As shown, our proposed method outperforms the baselines on all types of corruptions in both CIFAR10-C and CIFAR100-C datasets by a large margin. Additionally, we conduct an experiment on the CIFAR10-C dataset with the ResNet-26 architecture to make further comparisons of CAFA with other test-time adaptation methods. Note that MEMO [58] and TTT [49] adopt ResNet-26 with group normalization layers as their base architecture. As reported in Table 2, CAFA achieves the lowest error among the methods sharing the same pretrained model (*i.e.,* ResNet-26 (BatchNorm)). Furthermore, CAFA achieves the largest performance gain over the source model compared to MEMO [58] and TTT [49].

We further evaluate our method on a larger dataset, TinyImageNet-C, as shown in Table 3. Similar to the results of CIFAR10-C and CIFAR100-C, CAFA outperforms the baselines in all types of corruptions except for the contrast corruption on the TinyImageNet-C dataset. Even in such a case, the increased error is 0.1% which is marginal considering the performance gains in other corruption types. Finally, we validate our method on the most challenging corruption dataset, ImageNet-C. As reported in Table 4, even though our method is not a top performer, it outperforms most of the baselines by a large margin, which is 26.8% over the source model and 2.1% over TENT.

In the CIFAR10-C and CIFAR100-C datasets, CAFA achieves lower error rates compared to the TFA-Online and FR-Online methods which align source and target distributions without considering the class information. Such results demonstrate that considering both intra- and inter-class distances is important when performing feature alignments for test-time adaptation. Note that TFA-Online and TTT++-Online methods are online adaptation methods based on the original TFA and TTT++ [33] algorithms. Those are designed for offline adaptation that iterates multiple epochs over the entire set of test samples and predicts the test samples at once after multiple epochs. In their offline adaptation setting, TFA and TTT++ achieve 11.87% and 9.60% error rates on the CIFAR10-C dataset.[3]

4.2. Domain Adaptation beyond Image Corruptions

This section presents the experimental results for domain adaptation datasets beyond image corruption.

Datasets We adopt Office-Home [52] and DomainNet [39] datasets, which are widely used domain adaptation datasets. Office-Home [52] dataset consists of around 15,500 images and contains 65 categories of everyday objects with four

[3]Those numbers are obtained from Liu *et al.* [33]

Method	Ar→Cl	Ar→Pr	Ar→Re	Cl→Ar	Cl→Pr	Cl→Re	Pr→Ar	Pr→Cl	Pr→Re	Re→Ar	Re→Cl	Re→Pr	Average
Source	67.33	46.68	36.81	68.52	54.22	53.91	66.25	71.25	40.76	45.16	65.02	29.65	53.80
BN	63.34	47.31	36.29	66.09	57.85	54.12	62.22	68.93	39.29	46.27	60.73	30.19	52.72
PL	63.18	46.25	35.87	65.68	56.14	53.16	61.80	68.34	38.51	45.45	60.53	29.62	52.04
TENT†	61.47	44.33	34.82	62.75	52.22	49.16	61.60	66.19	36.26	44.66	58.08	28.14	49.97
T3A	62.29	**41.41**	34.11	64.65	51.05	48.91	60.61	66.96	**35.69**	45.69	59.22	**27.64**	49.85
AdaContrast	61.97	41.95	34.59	62.46	50.96	49.71	**59.21**	65.27	36.93	46.11	56.13	28.23	49.46
EATA†	62.86	43.64	34.43	63.37	**50.91**	48.70	60.77	65.91	35.99	43.30	56.40	27.84	49.43
CAFA (Ours)	**59.73**	42.64	**34.01**	**61.39**	51.23	**47.69**	60.28	**63.92**	35.87	**42.89**	**54.91**	27.84	**48.53**

Table 5. Classification error (%) on the OfficeHome [52] dataset. † denotes the results obtained from the official codes.

Method	Clip.	Info.	Pain.	Quic.	Real	Sket.	Avg.
Source	76.53	75.38	74.29	96.69	73.16	75.18	78.54
BN	76.14	79.25	72.90	93.15	74.17	68.80	77.40
PL	75.36	78.02	72.47	93.01	73.21	68.17	76.71
TENT†	84.59	75.66	71.81	92.86	71.60	67.85	75.73
T3A	74.10	76.50	71.68	92.89	73.50	66.64	75.88
EATA†	73.88	75.26	71.16	92.37	70.69	66.86	75.04
CAFA (Ours)	**73.17**	**74.69**	**71.05**	**92.49**	**69.96**	**66.53**	**74.65**

Table 6. Classification error (%) on the DomainNet [39] dataset. † denotes the results obtained from the official codes.

distinct domains: Artistic images (Ar), Clip art images (Cl), Product images (Pr), and Real-world images (Re). DomainNet [39] dataset is the largest domain adaptation dataset containing around 0.6 million images of 345 categories on six different domains which are clipart, infograph, painting, quickdraw, real, and sketch.

Quantitative evaluation and comparisons We evaluate our method on 12 different adaptation scenarios of the Office-Home [52] dataset, pretraining a model on one source domain and adapting it to the other domains. Table 5 shows the image classification errors (%) on different adaptation scenarios of the OfficeHome [52] dataset. As shown, CAFA consistently outperforms the baselines by a large margin. Ours reduces the classification error of the source model by around 5.3%, and that of TENT and EATA by around 1.4% and 0.9%, respectively.

Moreover, we also make comparisons with baselines on the DomainNet dataset, which is the largest domain adaptation dataset. For the experiment, we follow a similar evaluation protocol to the Office-Home dataset, *i.e.,* pretraining a model on each source domain and adapting the pretrained model to the other five domains. Each column denotes the source domain, and the numbers are the averaged classification error of the other five domains. As shown in Table 6, CAFA outperforms the existing methods by 1.1% and 0.4% over TENT and EATA, respectively.

4.3. Analysis

Pre-calculation of source statistics As aforementioned, CAFA requires access to the training samples to compute source statistics before adaptation. We demonstrate that we can obtain robust statistics for adaptation even with a small number of training samples, outperforming the existing methods. In Fig. 3, the red lines denote the performance

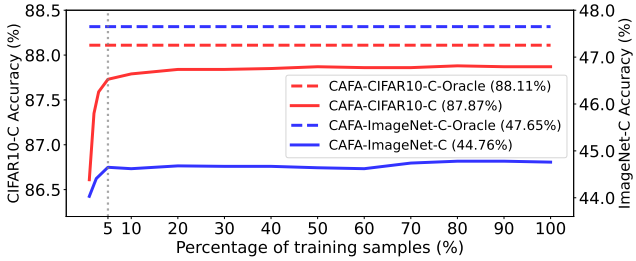

Figure 3. The test-time adaptation performance of CAFA on the CIFAR10-C and ImageNet-C datasets along with the percentage of train samples for calculating the train statistics.

of CAFA (solid line) and CAFA-Oracle (dotted line) on the CIFAR10-C dataset, and the blue lines indicate the performance on the ImageNet-C dataset. Note that oracle performances (*i.e.,* applying CAFA with ground truth labels) are measured using all training samples for obtaining statistics. As shown, around 5% of the training samples are sufficient to pre-calculate robust source statistics for achieving high adaptation performance in both CIFAR10-C and ImageNet-C datasets.

Effectiveness of Intra-/Inter-Class Dist.		Updating Batch Norm. vs Full Parameters		Tied vs Class-wise Covariance	
Source	29.14	Source	29.14	Source	29.14
Global FA	19.12	CAFA-Full	12.66	CAFA-Tied	12.47
Intra-Class Dist.	13.02	**CAFA**	**12.13**	**CAFA**	**12.13**
CAFA	**12.13**				

Table 7. Our ablation results on the CIFAR10-C dataset.

	Intra-/Inter-Class		BN vs Full Param.		Tied vs Class-wise Cov.	
CIFAR10-C	Global FA	19.12	Source	29.14	Source	29.14
	Intra-Class	13.02	CAFA-Full	12.66	CAFA-Tied	12.47
	CAFA	**12.13**	**CAFA**	**12.13**	**CAFA**	**12.13**
ImageNet-C	Global FA	72.38	Source	81.99	Source	81.99
	Intra-Class	59.12	**CAFA-Full**	**55.24**	CAFA-Tied	56.00
	CAFA	**55.24**	**CAFA**	**55.24**	**CAFA**	**55.24**
Office-Home	Global FA	55.35	Source	53.80	Source	53.80
	Intra-Class	49.50	CAFA-Full	48.47	CAFA-Tied	48.53
	CAFA	**48.27**	**CAFA**	**48.27**	**CAFA**	**48.27**

Table 8. Our ablation results on the CIFAR10-C, ImagNet-C, and Office-Home datasets.

Effectiveness of Intra-/Inter-Class Distance To further validate our motivation for considering intra- and inter-class distances, we conduct ablation studies on the CIFAR10-C, ImageNet-C, and Office-Home datasets. As reported in the

left group of Table 8, Global FA performs poorly in TTA since it does not consider the class information. In the case of reducing the intra-class distance only, it improves the classification errors over the source model, which is also effective. However, considering both intra- and inter-class distances (CAFA) achieves the lowest classification errors. Such results demonstrate our initial intuition is valid, which is elaborated on in Section 1. Note that we obtain consistent results on the CIFAR100-C dataset, as presented in the supplementary material.

Updating the entire parameters of feature extractor In our main experiments, we only update the modulation parameters β, γ of batch normalization layers in the networks, following Wang *et al.* [54]. In this ablation study, we further validate CAFA by updating the entire parameters of the feature extractor. Note that the classifier $h(\cdot)$ cannot be updated by our loss since CAFA performs alignments at a feature level. CAFA-Full in the middle group of Table 8 shows the result of updating the full parameters when utilizing CAFA. While it shows superior performance compared to the baselines, updating the batch normalization layers outperforms the case of updating the full parameters except for the ImageNet-C dataset. In the case of the ImageNet-C dataset, no difference in their performance is observed. As pointed out in Wang *et al.* [54], updating the full model may cause the model to diverge from what they learned from training. Furthermore, we conjecture that the number of samples during test-time adaptation may be not sufficient to optimize the entire parameters to converge.

Impact of using tied covariance We adopt the class-wise covariance matrices for CAFA in the main experiments. However, in Gaussian discriminant analysis (GDA), it is assumed that all the class-conditional Gaussian distributions share the same covariance matrix (*i.e.,* tied covariance). We conduct an ablation study regarding such an issue in the right group of Table 8. We observe that CAFA with tied covariance shows less performance improvement than utilizing the class-wise covariances. We conjecture that it is because class-wise covariances represent the statistics of each class of the source distributions more precisely than the tied covariance. Moreover, as pointed out in Lee *et al.* [28], deep networks are not trained to share the same covariance matrix for all class-conditional distributions. However, regardless of the covariance types, our method still outperforms the existing baselines.

Effectiveness of Mahalanobis distance To provide concrete background on the choice of Mahalanobis distance, we conduct experiments along with different distance or divergence types as in Table 9. As shown in the CIFAR10-C experiments, CAFA achieves the best accuracy while class-aware KL-divergence also improves the source performance by a large margin. However, it is not applicable to the ImageNet-C dataset since each test batch does not have enough samples to estimate the distribution for each class. On the other hand, Mahalanobis distance robustly outperforms other types since it measures the distance between a distribution and a sample.

Alignment types	Methods	CIFAR10-C	ImageNet-C
Class agnostic	Global FA	19.12	72.38
	KL-divergence	13.23	N/A
Class aware	KL-divergence	12.53	N/A
	Euclidean	13.20	56.30
	CAFA	**12.13**	**55.24**

Table 9. Ablation study on different distance types. Note that Euclidean distance is a special case of Mahalanobis distance.

Conditions		TENT†	CAFA (Ours)
Batch Size	BS $= 8$	14.47	13.02
	BS $= 4$	14.52	13.05
	BS $= 2$	15.10	13.43
	BS $= 1$	17.66	15.00
Source		29.14	
Label Shifts	$s = 0.1$	14.92	13.32
	$s = 0.3$	15.80	13.40
	$s = 0.5$	17.15	13.83
	$s = 0.7$	19.68	14.73
	$s = 0.9$	23.70	16.47
	$s = 1.0$	26.01	17.61
Source		29.14	

Table 10. Classification error (%) on the CIFAR10-C dataset with severity level 5 corruptions with small batch sizes (upper group) and label shifts (lower group).

Impact of small batch sizes and label shifts To show the wide applicability of CAFA to various deployment scenarios, we conduct ablation studies on 1) small batch sizes and 2) label shifts. In the experiment on small batch sizes ($<$ 10), we show that test-time adaptation approaches that assume a batch of test instances can still be effective in such scenarios, even when the batch size is equal to 1. To tackle this challenge, we modify how we compute the batch statistics. As pointed out in Schneider *et al.* [45], an effective solution is inferring test batch normalization statistics by leveraging the training batch statistics. Based on such intuition, we apply Schneider *et al.* [45] to CAFA and TENT and measure the performance with batch sizes less than 10. As reported in the upper group of Table 10, CAFA and TENT maintain reasonable performance even when the batch size equals 1, improving the source model by 14.1% and 11.5% respectively.

Additionally, we conduct an experiment assuming the label distribution of each test batch changes (*i.e.,* label shifts). To simulate the label shift, we sample the test instances from the multinomial distribution and change the distribution for each batch. With the given severity s and the number of classes c, we change the multinomial distribution from $[s + \frac{1-s}{c}, \frac{1-s}{c}, ..., \frac{1-s}{c}]$ to $[\frac{1-s}{c}, \frac{1-s}{c}, ..., s + \frac{1-s}{c}]$, where each element of the array is the probability of the

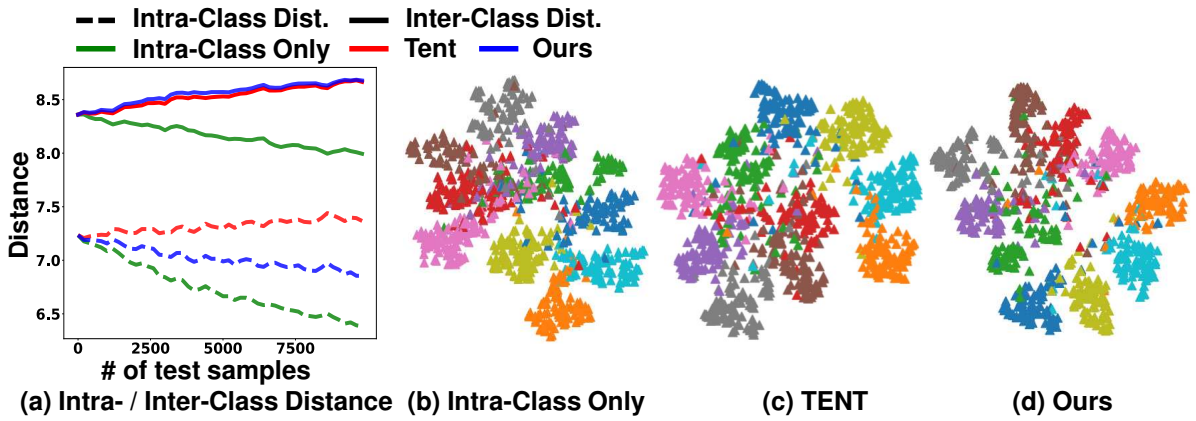

Figure 4. (a) illustrates the change of intra-class distance (dotted lines) and inter-class distance (solid line) as adaptation proceeds. (b-d) show the t-SNE visualizations of applying (b) intra-class distance only, (c) TENT [54], and (d) ours. All visualizations are obtained from the Gaussian noise corruption in the CIFAR10-C dataset.

corresponding class being sampled. For example, when the severity $s = 1$, the multinomial distribution changes from $[1, 0, 0, ..., 0]$ to $[0, 0, 0, ..., 1]$. Based on the same intuition from the experiment regarding the batch sizes, we apply Schneider *et al.* [45] to CAFA and TENT methods. As shown in the lower group of Table 10, CAFA maintains reasonable performance even when the label shift happens at the most by improving the source model by 11.5%. TENT also improves the source model under all label shift severities, but the performance gains are less than CAFA. We conjecture that this is because CAFA takes benefits of the feature alignment. Adapting a model without alignments may result in divergence from the source distribution and may collapse to trivial solutions under severe label shifts. On the other hand, in the case of CAFA, it aligns the target features to the source class-conditional distributions, which is more robust to such divergence.

Visualizations To further validate our intuition, we visualize the change of the intra- and inter-class distances in Fig. 4 (a): intra-class distance only (green line), TENT [54] (red line), and CAFA (blue line). As shown in Fig. 4 (a), reducing the intra-class distance alone also decreases the inter-class distance, which harms the class-discriminability. In the case of TENT [54], we observe that the intra-class distance does not decrease. On the other hand, CAFA desirably reduces the intra-class distance while enlarging the inter-class distance. Such improved class-discriminability is also shown in the t-SNE visualization in Fig. 4 (d). Ours shows more well-separated representation space in a class-wise manner compared to other methods (Fig. 4 (b) and (c)).

5. Discussion

In this work, we proposed a *simple yet effective* feature alignment that considers both intra- and inter-class distances, noting the importance of considering them for test-time adaptation. Most of the existing feature alignments are generally conducted along with training on the source data, which allows a model to learn target distributions in a class-discriminative manner. However, in the case of test-time adaptation where access to the source data is prohibited during adaptation, a model does not have a chance to learn test features in such a manner. Our simple feature alignment that considers both intra- and inter-class distances effectively addresses such a challenge as shown in our analyses and extensive experiments. However, there still remains room for improvements: 1) better pseudo-labeling method and 2) better learnable parameters than BN layers. We hope our work inspires the following researchers to investigate more effective test-time adaptation methods.

Pseudo labeling One limitation of our work is that we resort to using the predicted labels when assigning a class to each test sample. This could be problematic in the early phase of adaptation since the model encounters samples from different distributions, achieving low classification accuracy. Along with our novel perspective to consider both intra- and inter-class distance, we believe that replacing such a pseudo-labeling approach with advanced methods would further boost the adaptation performance of CAFA.

Finding better learnable parameters than BN layers In our work, we only updated the modulation parameters in batch normalization layers. Despite its demonstrated effectiveness, batch normalization parameters only take less than 1% of the model parameters. If we can find better learnable model parameters, it may further improve the adaptation performance of our proposed loss CAFA.

6. Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A5A708390811 and 2022R1A2B5B02001913).

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 2

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006. 2

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016. 1, 2

[4] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 627–636, 2019. 1, 3

[5] 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. 3, 5

[6] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11580–11590, 2021. 1

[7] Sungha Choi, Seunghan Yang, Seokeon Choi, and Sungrack Yun. Improving test-time adaptation via shift-agnostic weight regularization and nearest source prototypes. In *European Conference on Computer Vision*, pages 440–458. Springer, 2022. 2

[8] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3819–3824. IEEE, 2018. 1

[9] 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. 6

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[11] 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. *arXiv preprint arXiv:2010.11929*, 2020. 1

[12] 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*, 2021. 2, 5

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 1, 3

[14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 1

[15] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848. PMLR, 2016. 1, 2

[16] Philip Haeusser, Thomas Frerix, Alexander Mordvintsev, and Daniel Cremers. Associative domain adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2765–2773, 2017. 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5

[18] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. 1, 6

[19] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018. 1, 2

[20] Xuefeng Hu, Gokhan Uzunbas, Sirius Chen, Rui Wang, Ashish Shah, Ram Nevatia, and Ser-Nam Lim. Mixnorm: Test-time adaptation through online normalization estimation. *arXiv preprint arXiv:2110.11478*, 2021. 2, 3

[21] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 1

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 3

[23] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34, 2021. 2, 5

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[26] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. The parable of google flu: traps in big data analysis. *science*, 343(6176):1203–1205, 2014. 1

[27] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6

[28] Dongha Lee, Sehun Yu, and Hwanjo Yu. Multi-class data description for out-of-distribution detection. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1362–1370, 2020. 4, 8

[29] Dong-Hyun Lee. 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. 2, 3, 5

[30] Jungsoo Lee, Jooyeol Yun, Sunghyun Park, Yonggyu Kim, and Jaegul Choo. Improving face recognition with large age gaps by learning to distinguish children. 2021. 1

[31] Shuang Li, Shiji Song, Gao Huang, Zhengming Ding, and Cheng Wu. Domain invariant and class discriminative feature learning for visual domain adaptation. *IEEE transactions on image processing*, 27(9):4260–4273, 2018. 1, 3

[32] 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*, pages 6028–6039. PMLR, 2020. 1

[33] 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, 2021. 1, 2, 5, 6

[34] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018. 1, 3

[35] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013. 3

[36] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017. 1, 2

[37] Prasanta Chandra Mahalanobis. On test and measures of group divergence: theoretical formulae. 1930. 2, 3, 4

[38] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022. 5

[39] 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*, pages 1406–1415, 2019. 6, 7

[40] Pedro O Pinheiro. Unsupervised domain adaptation with similarity learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8004–8013, 2018. 2

[41] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008. 1

[42] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997. PMLR, 2017. 1, 3

[43] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018. 2

[44] 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. 2, 3, 5

[45] 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. 8, 9

[46] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. *Advances in neural information processing systems*, 29, 2016. 3

[47] Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018. 1, 3

[48] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 1

[49] 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. 1, 2, 5, 6

[50] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 1, 3

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[52] 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*, pages 5018–5027, 2017. 6, 7

[53] Georg Volk, Stefan Müller, Alexander Von Bernuth, Dennis Hospach, and Oliver Bringmann. Towards robust cnn-based object detection through augmentation with synthetic rain variations. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 285–292. IEEE, 2019. 1

[54] 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. 2, 3, 5, 6, 8, 9

[55] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9010–9019, 2021. 1

[56] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International conference on machine learning*, pages 5423–5432. PMLR, 2018. 1

[57] Fuming You, Jingjing Li, and Zhou Zhao. Test-time batch statistics calibration for covariate shift. *arXiv preprint arXiv:2110.04065*, 2021. 2, 3

[58] Marvin Mengxin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. 2, 3, 5, 6

[59] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3801–3809, 2018. 3