Feature Augmentation based Test-Time Adaptation

Younggeol Cho* Youngrae Kim* Junho Yoon Seunghoon Hong Dongman Lee
KAIST
{rangewing, youngrae.kim, vpdtlrdl, seunghoon.hong, dlee}@kaist.ac.kr

Abstract

Test-time adaptation (TTA) allows a model to be adapted to an unseen domain without accessing the source data. Due to the nature of practical environments, TTA has a limited amount of data for adaptation. Recent TTA methods further restrict this by filtering input data for reliability, making the effective data size even smaller and limiting adaptation potential. To address this issue, We propose Feature Augmentation based Test-time Adaptation (FATA), a simple method that fully utilizes the limited amount of input data through feature augmentation. FATA employs Normalization Perturbation to augment features and adapts the model using the FATA loss, which makes the outputs of the augmented and original features similar. FATA is model-agnostic and can be seamlessly integrated into existing models without altering the model architecture. We demonstrate the effectiveness of FATA on various models and scenarios on ImageNet-C and Office-Home, validating its superiority in diverse real-world conditions. Code is available at https://github.com/RangeWING/FATA.

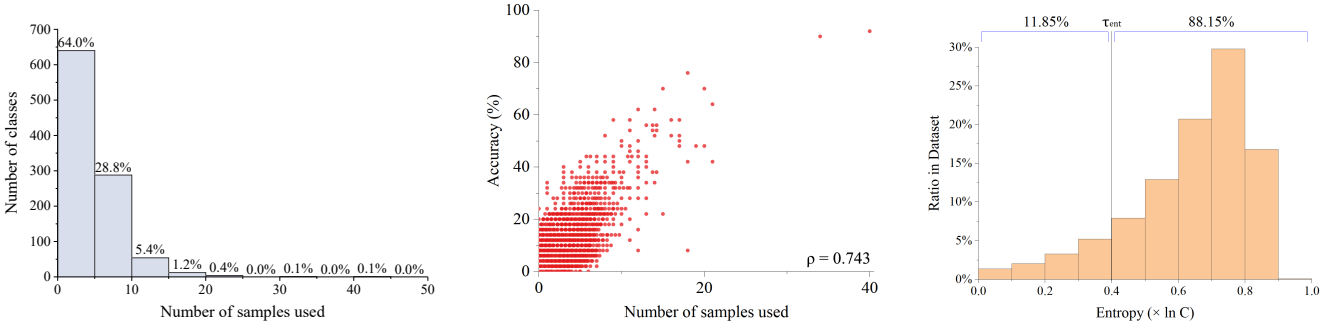
1. Introduction

Deep learning models have significantly enhanced the performance of computer vision applications [3, 8]. However, real-world scenarios often present challenges such as performance degradation caused by domain shifts between training and target domain. To mitigate this gap, unsupervised domain adaptation (UDA) [7, 17, 18, 21, 23, 26, 28] and test-time training (TTT) techniques [16, 27] have been proposed. These methods typically rely on adapting models to unseen domains using extensive source data during testing, which is often impractical due to limited computational resources and privacy issues. Recently, fully test-time adaptation (TTA) methods [1, 13, 19, 20, 33] have emerged, enabling online adaptation of trained models to target environments without the need for source data or labels. The dominant paradigm among TTA methods involves minimizing entropy loss while updating the affine parameters of Batch Normalization [11] layers, as initially proposed by TENT [33], which demonstrates the correlation between entropy and accuracy. Extending TENT, several methods propose sample selection based entropy minimization, which filters out unreliable or redundant samples. For example, Niu *et al.* [19] demonstrates that not all data samples are reliable and performs sample selection based on entropy and sample weighting based on the reliability for each sample. Similarly, SAR [20] filters out samples with high entropy. DeYO [13] also uses entropy for sample selection and filters out harmful samples that degrades the adaptation process, using structure or shape in the data for further sampling.

However, the limited amount of sampled data limits the performance improvement. For instance, only 11.85% of data from ImageNet-C [9] is selected and utilized by DeYO to perform naive entropy minimization [13], highlighting the inefficiency in leveraging the available samples. Furthermore, as depicted in Fig. 1a, 64.0% of the total classes are sampled less than five times, which leads to poor performance on those classes, as shown in Fig. 1b. Wang *et al.* [34] addresses this issue by using consistency loss, which involves comparing predictions with pseudo-labels predicted on augmented images for all samples. While this approach can mitigate the problem, it requires tens of inferences for each sample, rendering it impractical for real-world applications due to its high computational cost.

To fully utilize limited samples obtained by entropy-based sample selection, we propose a simple yet effective TTA method, named Feature Augmentation based Test-time Adaptation (FATA). FATA trains a model by comparing the pseudo-labels of reliable samples, obtained through entropy-based sample selection, with predictions made on the augmented features of these samples using normalization perturbation techniques [5, 14]. Augmented features allow the model to obtain the effect of having more reliable samples with only a small amount of data, thereby acquiring a more generalized representation. To augment the features, we adopt Normalization Perturbation (NP) [5], which randomly perturbs the features in a channel-adaptive man-

*These authors contributed equally.



(a) Count of classes for number of selected samples

(b) Accuracy of classes for number of samples used

(c) Entropy of samples in ImageNet-C

Figure 1. Problem analysis. We use ImageNet pretrained ResNet-50 [8] and Gaussian noise of level 5 from ImageNet-C [9]. We follow EATA [19] and SAR [20] to set the entropy threshold. (a) 64.0% of the classes are selected 5 times or fewer, where each class contains 50 images. (b) The less frequently a class is selected, the lower the performance. (c) Only 11.85% of samples in ImageNet-C are used when entropy based filtering is used.

ner using channel statistics. This enables greater variation in features, not limited to the small variance of the source domain, and facilitates learning from a more diverse set of samples. Following Stochastic Feature Augmentation [14], this mechanism is embedded between the last two blocks of the backbone model, mitigating the inefficiency of multiple inferences since only two layers of the model are involved in the process of prediction. FATA can be seamlessly integrated into any method using entropy-based sample selection techniques and is applicable to any architecture.

To validate the effectiveness of FATA, we evaluate our method not only in normal scenarios but also in several challenging scenarios that are likely to occur in the real world, such as label shifts and batch size 1 scenarios, following the settings in SAR [20]. In our evaluation, the methods incorporating FATA demonstrate superiority in performance, showcasing applicability to various methods and network architectures. Additionally, our meticulously designed ablation studies and analysis illustrate the effectiveness of our components.

Our contributions are summarized as follows:

- We analyze and address the problem of data scarcity in sample selection based TTA methods. We observe that the majority of data is filtered out and that the number of samples used has a positive correlation with the performance.
- We propose Feature Augmentation based Test-time Adaptation (FATA), a TTA method that fully exploits limited amount of data. FATA leverages feature augmentation and augmentation loss, and can be seamlessly plugged into any model or TTA method.
- We validate FATA on several models and scenarios and demonstrate its effectiveness. FATA outperforms existing methods when plugged into the methods.

2. Related Work

2.1. Test-Time Adaptation

In order to enable model adaptation in source-free, unlabeled, and online settings, various test-time domain adaptation methodologies [1,12,15,20,20,22,33,35,36] have been introduced, designing unsupervised losses. TTA methodologies can be broadly categorized into two approaches: one that utilizes entropy minimization and another that generates reliable pseudo-labels through multiple data augmentation to improve the prediction accuracy.

TENT [33] is the first approach to highlight the test-time adaptation of pre-trained models to given target samples by employing entropy minimization loss. Followed by TENT, several methods [13,19,20] employ the entropy minimization. EATA [19] suggests sample filtering for reliable adaptation. The authors found that test samples with high entropy lead to noisy gradients, resulting in a severe performance drop. Consequently, the authors filters out samples with high entropy. SAR [20] proposes sharpness-aware and reliable entropy minimization to address the problem of real-world scenarios, such as small batch sizes and online imbalanced label shifts. SAR identifies that samples with large gradient norm also hinder the adaptation process, even if their entropy is low. Based on these observations, SAR minimizes both the sharpness of the entropy loss and the entropy itself, using SAM optimizer [6]. DeYO [13] observed that using entropy alone as a sample selection criterion is insufficient, as it does not account for the discriminability of the sampled data, such as structure or shape. The authors demonstrate that samples without the discriminability can be harmful for the adaptation process, even if thy have low entropy. To address this problem, DeYO further filters out non-discriminative data from the low entropy samples, using their proposed metric. Although those sample selection

based entropy minimization methods have shown promising results, they are limited in performance improvement because they do not utilize the majority of target samples.

On the other hand, CoTTA [34] augments an input data and performs tens of inferences to generate reliable pseudo-labels, which are then used to train the model. While this method utilizes all the data and addresses the inefficiency problem of entropy minimization-based methods, it is impractical in the real-world applications due to its heavy computational burden.

2.2. Feature Augmentation

In the field of domain generalization, data augmentation has been demonstrated to be an effective method for fully leveraging source data. Existing data augmentation methods [24, 32, 37] rely on image-space operations, which require careful augmentation design and substantial computational resources. Recently, feature augmentation has been proposed as a solution to address the limited diversity and inefficiency of data augmentation [14, 31, 38]. By applying transformations in the feature space [14] to simulate various feature distributions during training [29], feature augmentation enhances model generalization to new domains more effectively than traditional data augmentation methods.

MixStyle [38] proposes an explicit augmentation approach that perturbs latent features using domain labels through interpolation. Similarly, Li *et al.* (2021) [14] presents Stochastic Feature Augmentation (SFA), which augments the latent features using a linear function with randomly sampled weights and biases from normal distributions. SFA can be implemented as a plug-in module, making it adaptable for integration into various models.

Meanwhile, Fan *et al.* (2023) [5] propose Normalization Perturbation (NP) inpired by Adaptive Instance Normalization (AdaIN) [10], a style transfer method that utilizes normalization and transformation of feature channel statistics. Instead of directly perturbing features, NP perturbs the channel statistics to effectively maintain feature content. While feature augmentation techniques have proven effective for domain generalization, their impact on TTA has not been thoroughly explored. Our method integrates feature augmentation into TTA to addresses the data scarcity issues by sampling data based on entropy values.

3. Problem Analysis

3.1. Preliminaries

Test-Time Adaptation. Unsupervised domain adaptation (UDA) adapts a model to a target domain without target labels. Source-free domain adaptation removes the necessity of source data from UDA. Test-time training (TTT) adds the constraint of online availability of target data. TTT addresses this problem by also training the model online, i.e., in the test time, considering that the target data could be acquired from the real world, in real time. Finally, fully test-time adaptation addresses the most realistic scenario where no source data is available. TTA only adapts a model with unlabeled target data in the test-time.

Entropy Minimization. In TTA scenario, we have a model f_θ with parameter θ pretrained on the source dataset $\mathcal{D}^{\text{train}} = \{(\mathbf{x}_i^{\text{train}}, \mathbf{y}_i^{\text{train}})\}_{i=1}^{N^{\text{train}}}$. Due to the absence of $\mathcal{D}^{\text{train}}$ and labels $\mathbf{y}^{\text{test}}$ of the target dataset $\mathcal{D}^{\text{test}} = \{(\mathbf{x}_i^{\text{test}})\}_{i=1}^{N^{\text{test}}}$ in test time, existing TTA methods have employed unsupervised learning signal. The most dominantly adopted learning signal is Shannon entropy [25], where the model predicts to minimize the entropy of prediction. Given a model f_θ and a prediction $\hat{y}_i = f_\theta(c|\mathbf{x})$ according to a class c, entropy minimization is formulated as follows:

$$\min \text{Ent}_\theta(\mathbf{x}), \text{ where } \text{Ent}_\theta(\mathbf{x}) = -\sum_{i=1}^{C} \hat{y}_i \log \hat{y}_i, \quad (1)$$

where C is the number of classes.

Sample Selection Strategy. Based on entropy minimization, Niu *et al.* [19] proposes filtering out unreliable samples by using only samples with low entropy values. Therefore, the model minimizes entropy using the samples selected based on sample selection criteria $S(\mathbf{x})$:

$$\min_\theta S(\mathbf{x})\text{Ent}_\theta(\mathbf{x}), \quad \text{where } S(\mathbf{x}) \triangleq \mathbb{I}_{\{\mathbf{x}\in\mathcal{S}\}}, \quad (2)$$

where $\mathbb{I}_{\{\cdot\}}(\cdot)$ is an indicator function and $\mathcal{S}$ is a set of selected samples. For instance, the set for entropy-based sample selection is $\mathcal{S}_{\text{ent}} = \{\mathbf{x}|\text{Ent}_\theta(\mathbf{x}) < \tau_{ent}\}$, where τ_{ent} is a pre-defined entropy threshold.

3.2. Analysis on Sample Selection Strategy

We measure how many samples are selected by this sampling strategy. We use ResNet50-BN (Batch Normalization) pretrained on ImageNet [2], and count the selected numbers from target dataset $\mathcal{D}^{\text{test}}$, ImageNet-C [9], with the entropy threshold τ_{ent} that the authors of EATA and SAR [20] recommended.

Use of a Limited Number of Samples. Limited samples could degrade the performance of adaptation in the real world when the online target data is small or there is a label imbalance. As shown in Fig. 1c, only 11.85% of $\mathcal{D}^{\text{test}}$ can be used when the entropy-based filtering strategy is used. This is very small number regarding the number of classes in the dataset, a thousand. This indicates that only five or six samples can be used for adaptation for each class on average (see Fig. 1a), where the accuracies of the less sampled classes are low (see Fig. 1b), leading to poor performance. Additionally, DeYO [13] selects samples that have helpful

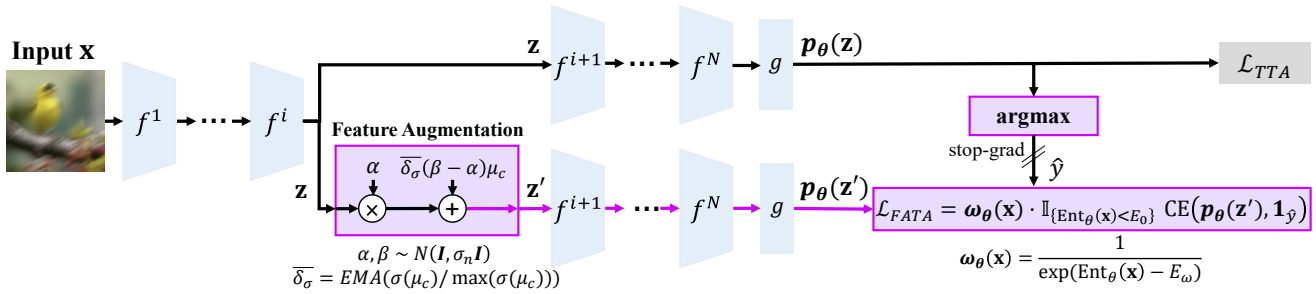


Figure 2. Overview of FATA. There are two prediction branches where one is for obtaining pseudo-label on the reliable data and another is for prediction and updating the model on the augmented feature. We insert the feature augmentation after the i-th layer.

structure and shape among the selected sample by entropy-based filtering, resulting in usage of much less samples.

Despite the limited amount of sampled data, these methods naively rely solely on entropy loss without considering generalized representation. We conjecture that this would restrict the exposure to the target domain, leading to limited performance improvement. To enhance the model's exposure to the target domain by further exploiting reliable sampled data, a more sophisticated method is necessary.

4. Method

4.1. Feature Augmentation

We augment the target samples to fully exploit the limited amount of data, as shown in Fig. 2. Instead of conventional data augmentation, we adopt feature augmentation [14], which allows diverse augmented features. Given an encoder f composed of N layers $f^1, f^2, \cdots, f^N$ and a sample $\mathbf{x} \in \mathbb{R}^{B\times C\times H\times W}$, feature augmentation augments an intermediate feature from the i-th layer, $\mathbf{z} = f^i \circ f^{i-1} \circ \cdots \circ f^1(\mathbf{x})$. For example, Normalization Perturbation Plus (NP+) [5] perturbs the channel statistics of an intermediate feature $\mathbf{z} \in \mathbb{R}^{B\times C_i\times H_i\times W_i}$ using normalization for domain generalization as follows:

$$\mathbf{z}' = \alpha\mathbf{z} + \delta(\beta - \alpha)\mu_c, \tag{3}$$

where α, $\beta \in \mathbb{R}^{B\times C}$ are the random noises sampled from $N(\mathbf{I}, \sigma_n\mathbf{I})$, $\delta = \mathrm{Var}(\mu_c)/\max(\mathrm{Var}(\mu_c))$ is the normalized variance, $\mu_c = \{\mu_c^j\}_{j=1}^B \in \mathbb{R}^{1\times C_i}$ is the channel-wise feature mean.

FATA augments an intermediate feature $\mathbf{z}$ as follows:

$$\mathbf{z}' = \alpha\mathbf{z} + \overline{\delta_\sigma}(\beta - \alpha)\mu_c, \tag{4}$$

where $\overline{\delta_\sigma}$ is the exponential moving average of the normalized standard deviation $\delta_\sigma = \sigma(\mu_c)/\max(\sigma(\mu_c))$ and σ is the standard deviation operator. We replace δ with $\overline{\delta_\sigma}$ to address the following issues. Firstly, unlike domain generalization, TTA adapts to a specific domain from a limited amount of data. To adaptively adjust the noise to fit the target domain, we add an exponential moving average that estimates the statistics of the target domain. Secondly, Eq. (3) introduces the variance with the magnitude of square of variance, as the normalized statistic variance δ adjusts the channel mean μ_c to control the random noise for each channel. To address this issue, we replace the variance to the standard deviation.

4.2. FATA Loss

In order to fully utilize the augmented features, we propose the FATA loss, an augmentation loss that is applied to augmented features.

Augmentation Loss. Given a classifier g, the output probability $\mathbf{p}_\theta(\mathbf{z}) = g \circ f^N \circ f^{N-1} \circ \cdots \circ f^{i+1}(\mathbf{z})$. We propose an augmentation loss based on cross-entropy between an output of the augmented feature and a pseudo-label from the original feature. Given the pseudo-label $\hat{y} = \mathrm{stopgrad}(\mathrm{argmax}(\mathbf{p}_\theta(\mathbf{z})))$, our augmentation loss is formulated as follows:

$$\mathcal{L}_{\mathrm{aug}}(\mathbf{x};\theta) = \mathrm{CE}(\mathbf{p}_\theta(\mathbf{z}'), \mathbf{1}_{\hat{y}}). \tag{5}$$

Unlike CoTTA [34], FATA updates a model on the augmented features and uses a prediction on the original data as a pseudo-label. Therefore, the model can make predictions on more diverse features and be updated on those features. Also, the pseudo-label is reliable because the data has already been sampled by the entropy threshold E_0.

Sample Selection and Weighting. Following EATA [19], we apply entropy-based sample selection and sample weighting. Given an entropy threshold E_0 and a normalization factor E_w, the sample selection criteria is $\{\mathbf{x}|\mathrm{Ent}_\theta(\mathbf{x}) < E_0\}$ and the sample weighting function ω_θ is formulated as $\omega_\theta(\mathbf{x}) = 1/\exp\left(\mathrm{Ent}_\theta(\mathbf{x}) - E_w\right)$.

FATA Augmentation Loss. Finally, we incorporate sample selection and weighting to our augmentation loss as follows:

$$\mathcal{L}_{\mathrm{FATA}}(\mathbf{x};\theta) = \omega_\theta(\mathbf{x}) \cdot \mathbb{I}_{\{\mathrm{Ent}_\theta(\mathbf{x})<E_0\}}\mathrm{CE}(\mathbf{p}_\theta(\mathbf{z}'), \mathbf{1}_{\hat{y}}), \tag{6}$$

where $\mathrm{CE}(p, q)$ is the cross-entropy function.

Method	Noise			Blur				Weather				Digital				Avg.	ΔPerf.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG		
No Adapt	16.06	16.76	16.72	14.94	15.36	26.23	38.87	34.29	33.21	47.75	65.35	16.91	43.97	49.04	40.01	31.70	-
TENT [33]	29.36	31.30	30.13	28.22	27.88	41.40	49.30	47.36	41.73	57.53	67.50	30.08	54.90	58.58	52.57	43.19	-
EATA [19]	34.58	37.34	35.87	33.55	33.24	47.40	52.90	51.72	45.71	59.86	68.12	44.65	57.90	60.39	55.03	47.88	-
EATA+FATA	35.31	38.00	36.31	34.63	34.32	48.15	52.44	52.17	46.26	59.93	67.43	46.48	57.77	60.19	55.05	48.29	+0.41
SAR [20]	29.99	31.98	30.95	28.33	26.11	42.01	49.51	47.63	42.61	57.70	67.37	39.46	54.58	58.62	52.64	43.97	-
SAR+FATA	35.00	37.05	35.70	33.55	32.61	47.35	51.55	51.23	45.29	59.33	67.08	42.23	57.30	60.06	54.72	47.34	+3.40
DeYO [13]	35.68	38.19	37.39	33.99	33.65	48.27	52.94	52.34	46.32	60.50	68.01	44.34	58.25	61.16	55.58	48.44	-
DeYO+FATA	37.06	38.76	38.05	34.80	34.59	49.16	52.91	52.82	46.78	60.67	67.73	47.67	58.47	61.19	55.69	49.09	+0.65

(a) ResNet50 (BN)

Method	Noise			Blur				Weather				Digital				Avg.	ΔPerf.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG		
No Adapt	17.98	19.83	17.87	19.75	11.35	21.41	24.91	40.43	47.31	33.60	69.28	36.27	18.61	28.40	52.28	30.62	-
TENT [33]	6.45	9.12	8.11	16.62	12.66	25.54	28.93	31.87	39.63	4.50	71.11	43.81	15.81	49.57	55.59	27.95	-
EATA [19]	37.16	40.33	38.64	28.98	27.58	36.69	38.72	51.25	49.47	55.12	71.98	49.77	41.39	55.90	57.90	45.39	-
EATA+FATA	42.67	45.64	43.92	32.53	31.07	42.96	46.72	57.03	54.81	61.80	73.81	54.56	51.12	60.94	60.89	50.70	+5.31
SAR [20]	28.56	30.96	29.85	18.39	18.15	30.67	30.70	41.97	43.76	6.20	70.75	44.08	15.50	48.94	55.28	34.25	-
SAR+FATA	40.15	42.46	40.85	24.99	25.21	38.33	40.46	52.90	50.85	0.23	73.47	49.92	40.17	55.94	57.31	42.22	+7.97
DeYO [13]	39.46	41.90	41.03	22.27	24.11	38.48	37.87	50.51	49.59	1.43	73.17	49.95	41.54	55.96	57.82	41.67	-
DeYO+FATA	39.71	42.49	41.37	22.29	24.46	38.90	38.31	51.23	50.02	56.31	73.19	50.03	42.10	55.99	57.79	45.61	+3.97

(b) ResNet50 (GN)

Method	Noise			Blur				Weather				Digital				Avg.	ΔPerf.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG		
No Adapt	9.48	6.75	8.23	28.99	23.45	33.87	27.13	15.91	26.49	47.18	54.66	44.09	30.55	44.50	47.80	29.94	-
TENT [33]	42.59	1.92	43.80	52.49	48.43	55.72	51.01	30.03	24.92	66.67	74.90	64.66	53.72	67.00	64.27	49.48	-
EATA [19]	50.23	49.70	51.33	55.32	55.44	59.64	57.00	63.29	62.38	70.85	75.95	66.95	64.16	69.39	67.51	61.28	-
EATA+FATA	53.15	53.21	54.05	57.23	57.84	62.08	60.78	66.76	65.41	72.10	76.89	67.65	67.13	71.68	69.01	63.67	+2.39
SAR [20]	44.03	42.96	45.54	53.07	49.81	55.78	51.44	58.00	55.43	66.34	74.58	64.14	55.02	66.84	64.06	56.47	-
SAR+FATA	51.67	3.69	52.48	57.00	56.93	61.42	59.76	65.70	64.70	71.33	76.21	64.42	66.43	71.70	68.30	59.45	+2.98
DeYO [13]	48.56	47.66	53.66	58.32	58.53	63.05	59.96	67.12	65.85	73.23	77.97	67.98	67.87	73.17	69.90	63.52	-
DeYO+FATA	46.83	53.78	54.2	58.56	58.57	63.37	60.16	67.46	66.24	73.46	78.19	68.55	67.78	73.33	69.93	64.03	+0.51

(c) ViT-B (LN)

Table 1. Image classification results on ImageNet-C [9]. ResNet50 with BN/GN and ViT-B with LN are used for this experiment. We use the accuracy (%) as the metric. ΔPerf. is the performance gap between methods without and with FATA.

Total Loss. Given a TTA loss $\mathcal{L}_{\text{TTA}}$ such as entropy minimization loss, we incorporate the FATA augmentation loss to the TTA loss. Consequently, the total loss $\mathcal{L}$ is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{TTA}} + \mathcal{L}_{\text{FATA}}, \tag{7}$$

which combines TTA loss and FATA augmentation loss. The proposed loss can be plugged into any method, without modifying the TTA loss such as sample selection based entropy minimization loss.

5. Experiment

5.1. Experimental Settings

Benchmark and Test Scenarios. We use the ImageNet-C [9] and the Office-Home [30] datasets for our evaluation. ImageNet-C includes corrupted images from the ImageNet dataset across 15 different corruption types. The dataset contains 50,000 images for each corruption type, resulting in a total of 750,000 images. Office-Home includes 15,588 images of 65 classes from 4 domains. Following the scenarios outlined by Niu *et al.* [20], we validate the effectiveness and robustness of our method in three scenarios: 1) In the normal scenario, a model adapts to streaming corrupted input where the label distribution is balanced, allowing the use of a large batch size; 2) In the batch size of one scenario, a model is exposed to a single image per iteration; 3) In the online imbalanced label distribution shift scenario, the labels within a batch are highly imbalanced.

Models. To demonstrate the applicability of our method to various models, we use models incorporating three different normalization layers in our experiments: Batch Normalization (BN), Group Normalization (GN), and Layer Normalization (LN). For BN and GN, we use ResNet-50, and for LN, we use the ViT-Base model. For ImageNet-C, all the models are pretrained on ImageNet dataset and adapted at test-time. For Office-Home, the models are pretrained on a source domain and adapted to a target domain at test-time.

Baselines. We compare our method to state-of-the-art fully test-time adaptation methods, TENT [33], EATA [19], SAR [20], and DeYO [13]. Since our method is proposed to address data scarcity when sample filtering is used, we integrate our method with EATA, SAR, and DeYO, which all employ filtering strategies. We further compare our method

Method	Noise			Blur				Weather				Digital				Avg.	ΔPerf.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG		
No Adapt	17.98	19.84	17.88	19.75	11.35	21.42	24.92	40.43	47.30	33.59	69.28	36.27	18.61	28.40	52.28	30.62	-
TENT [33]	3.36	4.31	4.18	16.75	3.49	28.00	29.36	18.65	24.72	2.21	72.03	46.17	8.12	52.43	56.25	24.67	-
EATA [19]	24.68	28.01	25.53	17.75	17.43	28.68	29.20	44.52	44.34	41.92	70.93	44.86	27.53	45.90	55.62	36.46	-
EATA+FATA	26.17	31.42	27.12	20.34	17.09	32.20	23.94	49.54	50.02	11.05	72.86	49.64	7.40	51.85	58.38	35.27	-1.19
SAR [20]	23.35	26.27	23.82	18.71	15.54	28.78	30.62	45.55	44.93	25.58	72.18	44.56	15.04	47.22	56.05	34.55	-
SAR+FATA	34.91	39.20	36.45	24.14	22.33	36.89	39.47	54.26	51.55	8.06	73.88	50.98	41.29	55.74	58.54	41.85	+7.30
DeYO [13]	41.34	44.11	42.69	22.39	24.22	41.43	28.93	54.06	51.79	2.14	73.17	53.42	47.84	59.86	59.67	43.14	-
DeYO+FATA	42.06	44.52	42.52	26.75	27.33	42.48	43.31	56.42	54.07	2.58	73.97	54.10	48.28	60.21	60.39	45.27	+2.13

(a) ResNet50 (GN)

Method	Noise			Blur				Weather				Digital				Avg.	ΔPerf.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG		
No Adapt	9.48	6.75	8.23	28.99	23.45	33.87	27.13	15.91	26.49	47.18	54.66	44.09	30.55	44.50	47.80	29.94	-
TENT [33]	42.72	1.40	44.53	52.49	48.67	56.10	51.09	22.96	20.87	66.78	75.01	64.98	53.96	67.03	64.38	48.86	-
EATA [19]	33.54	25.16	31.98	44.70	38.86	47.11	42.37	39.67	40.81	61.14	67.81	61.79	47.51	59.00	59.18	46.71	-
EATA+FATA	39.33	1.29	37.47	48.06	43.99	51.20	47.29	48.75	47.02	66.32	71.84	64.09	56.50	63.51	62.94	49.97	+3.12
SAR [20]	41.29	36.39	42.08	53.54	50.68	57.62	52.78	58.68	44.52	68.61	75.90	65.46	57.86	68.79	65.91	56.01	-
SAR+FATA	50.28	48.57	51.20	57.08	56.69	61.94	59.65	65.79	64.58	71.64	76.67	66.78	66.40	72.06	68.13	62.50	+5.46
DeYO [13]	53.29	53.92	54.22	58.85	59.34	64.45	49.76	68.06	66.63	73.71	78.23	68.15	68.75	73.76	70.74	64.12	-
DeYO+FATA	52.87	52.89	53.87	58.34	58.60	63.84	61.03	67.56	65.91	72.66	77.49	67.64	68.15	72.96	69.81	64.24	+0.12

(b) ViT-B (LN)

Table 2. Image classification results on ImageNet-C [9] under batch size 1. ResNet50 with GN and ViT-B with LN are used for this experiment. We use the accuracy (%) as the metric. ΔPerf. is the performance gap between the original method and another version where our method is incorporated.

to MEMO [36] and CoTTA [34] on Office-Home.

Implementation Details. We implement FATA on the PyTorch framework. We set E_0 to $0.5 \ln C$ and E_ω to $0.4 \ln C$, following the setting of DeYO [13], where C is the number of classes in the target dataset. In addition, we use the same optimizer as the existing methods with which we integrate FATA. We set the layer to inject the feature augmentation i to 3 for ResNet50 and 11 for ViT-B. We set the standard deviation for the noise σ_n to 1.0 and the smoothing factor for exponential moving average λ_{EMA} to 0.95. We use the default batch size of 64. We set the learning rate as 0.0005 and 0.001 for ResNet50 and ViT-B, respectively. When the batch size is one, we set the learning rate as 0.00025 and 0.000016 for ResNet50 and ViT-B, respectively.

5.2. Results

Comparison on Normal Scenario. Tab. 1 shows the comparison of performance in the normal scenario. The results clearly indicate that integrating our proposed method, FATA, yields superior results across various architectures and methods, compared to the original versions. Notably, the enhancement is most significant when ResNet-50 with GN is utilized, showcasing an impressive average improvement of up to 7.97 points. This substantial gain highlights the effectiveness of FATA, demonstrating its capability regardless of the underlying model architecture or TTA method. Furthermore, the consistent improvements across different settings demonstrate the robustness of FATA.

Comparison under Batch Size of One. Tab. 2 depicts the comparison of the performance in the batch size one scenario. Overall, our proposed method aids the model in adapting more effectively than when the original methods are employed independently. While there is a case where our method exhibits a slight performance drop, the performance improvements generally surpass the amount of drop, indicating that our method remains effective, despite the challenging constraints of the batch size one scenario.

Comparison under Imbalanced Label Shifts. As shown in Tab. 3, our proposed method demonstrates its effectiveness in the online imbalanced label distribution shift scenario. Remarkably, when our method is combined with EATA, the accuracy improvement reaches an impressive peak of 17.12 points. This substantial enhancement is not limited to EATA alone; our approach also significantly boosts the performance of other methods.

Comparison on Office-Home. Tab. 4 shows the comparison of the performance on the Office-Home dataset. FATA consistently enhances the performance with small computational overhead compared to CoTTA, demonstrating the efficiency and robustness of our method.

5.3. Ablation study

Location Choice for Feature Augmentation. We present the ablation study on the position of feature augmentation in Tab. 5. Inserting the feature augmentation after the third layer achieves the best performance in average accuracy, therefore, we set $i = 3$ as the default. Embedding it after layer 2 shows comparable performance to that after layer 3.

Method	Noise			Blur				Weather				Digital				Avg.	ΔPerf.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG		
No Adapt	17.87	19.91	17.90	19.70	11.21	21.28	24.86	40.38	47.40	33.57	69.24	36.27	18.65	28.34	52.17	30.58	-
TENT [33]	3.54	4.17	3.78	16.73	7.53	25.73	31.50	20.14	29.51	2.39	72.17	46.09	8.38	52.11	56.22	25.33	-
EATA [19]	25.26	29.49	27.72	14.90	16.61	24.41	28.60	34.89	29.71	41.41	62.65	35.22	26.14	41.70	48.69	32.49	-
EATA+FATA	41.75	43.98	41.78	27.95	27.63	42.47	45.80	56.82	53.83	62.12	73.32	54.33	50.38	61.42	60.52	49.61	+17.12
SAR [20]	33.81	36.90	33.90	18.35	20.48	33.13	33.97	29.87	45.08	2.77	71.95	46.78	7.43	51.91	56.06	34.83	-
SAR+FATA	42.97	45.32	43.91	28.29	27.76	41.51	44.26	55.21	53.72	1.56	73.78	53.27	2.43	60.14	59.47	42.24	+7.41
DeYO [13]	40.84	18.13	3.46	22.63	23.64	41.10	7.11	52.57	51.39	58.72	73.12	52.21	46.38	59.09	59.43	40.66	-
DeYO+FATA	44.30	46.29	1.06	25.89	29.58	45.00	5.06	58.27	54.69	62.98	73.85	55.52	53.23	62.52	61.19	45.30	+4.64

Table 3. Image classification results on ImageNet-C [9] under imbalanced label distribution shifts scenario. ResNet50 with GN is used for this experiment. We use the accuracy (%) as the metric. ΔPerf. is the performance gap between the original method and another version where our method is incorporated.

Methods	No Adapt	MEMO [36]	TENT [33]	CoTTA [34]	EATA [19]	EATA+FATA	SAR [20]	SAR+FATA	DeYO [13]	DeYO+FATA
Accuracy (%)	58.35	58.15	58.36	57.57	58.58	59.71	58.37	59.18	58.56	58.64
GMACs	4.11	262.93	4.11	143.83	4.11	4.92	8.22	14.76	8.22	9.03

Table 4. Image classification results with computational complexity on Office-Home [30]. Accuracy is averaged across all domain shift scenarios. ResNet50 with GN is used.

Method	FATA Position	Avg.
No Adapt	-	30.62
DeYO [13]	-	41.67
DeYO+FATA	0	41.52
DeYO+FATA	1	42.32
DeYO+FATA	2	43.01
DeYO+FATA	3	**45.61**

Table 5. Ablation study on the position of feature augmentation. ResNet50 with GN is used for this experiment.

Method	EMA	Std. dev.	Avg.
NP+	✗	✗	44.84
	✓	✗	44.96
FATA (Ours)	✓	✓	**45.61**

Table 6. Ablation study on the component of feature augmentation. ResNet50 with GN and the default batch size of 64 are used.

However, for certain corruption types (e.g., Fog), it exhibits a tremendous performance drop, indicating instability. Although several works [4, 5] have argued that augmenting the features in the shallow layers is the most effective for modifying the style of the input, our results indicate that inserting augmentation after the first layer shows the lowest performance, similar to Li *et al.* (2021) [14].

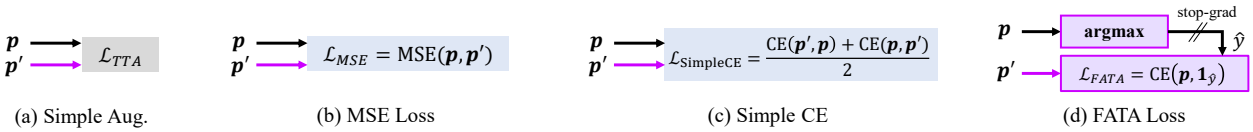
Component of Feature Augmentation. Tab. 6 shows the ablation study on the component of feature augmentation. Compared to NP+ [5], adding EMA and modifying the variance term to the standard deviation improve the accuracy, showing their effectiveness.

We ablate the augmentation loss in Tab. 7. As shown in Fig. 3a, Simple Augmentation (Simple Aug.) denotes using entropy instead of $\mathcal{L}_{\text{FATA}}$, i.e., it utilizes entropy loss on the output of augmented feature. MSE Loss denotes using mean square error loss between the outputs, as depicted in Fig. 3b. Simple CE denotes employing cross-entropy instead of $\mathcal{L}_{\text{FATA}}$, as shown in Fig. 3c. In detail, Simple CE excludes the process of obtaining pseudo-labels from our proposed method and uses cross-entropy loss between the output distribution of the augmented features and that of the original feature.

DeYO	Augmentation loss	Avg.
-	-	30.62
-	FATA loss (Ours)	39.35
✓	-	41.67
✓	Simple Aug.	40.30
✓	MSE Loss	6.54
✓	Simple CE	44.73
✓	FATA Loss (Ours)	**45.61**

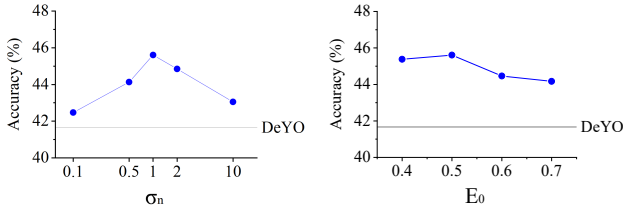
Table 7. Ablation study on the augmentation loss. ResNet50 with GN is used for this experiment.

Training a model solely with our method boosts performance by 8.74% in average accuracy, which is comparable to the performance of the state-of-the-art method, DeYO. Comparing our method with Simple Augmentation, it is evident that augmenting the features rather than the input data is much more effective, as it guides the model to have more generalized representations. In the case of Simple CE, the accuracy on all corruption types are almost zero. This is because naively comparing the output distribution of the original sample and augmented sample leads to the phenomenon of model collapse, as the model learns different features to map to the same output. To avoid this, we do not compare the distribution of the output and adopt pseudo-labeling techniques. In the end, incorporating DeYO and our method achieves the best performance.



(a) Simple Aug. (b) MSE Loss (c) Simple CE (d) FATA Loss

Figure 3. Augmentation losses for the ablation study. $\mathbf{p}$ and $\mathbf{p}'$ denotes $\mathbf{p}_\theta(\mathbf{z})$ and $\mathbf{p}_\theta(\mathbf{z}')$, respectively.



(a) Parameter sensitivity of σ_n. (b) Parameter sensitivity of E_0.

Figure 4. Hyperparameter Sensitivity.

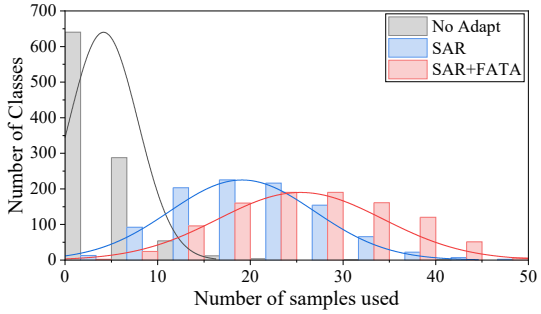


Figure 5. Count of classes for each number group of selected samples.

Hyperparameter Sensitivity. We demonstrate the sensitivity of the hyperparameters in Fig. 4. We conduct experiments to assess the sensitivity of two hyperparameters: the entropy threshold E_0 and the standard deviation for the noise σ_n. As shown in Fig. 4a, the highest accuracy is achieved with $\sigma_n = 1$, although other values, ranging from $\sigma_n = 0.1$ to 10, consistently outperform DeYO. Similarly, Fig. 4b shows that an entropy threshold of $E_0 = 0.5$ yields the best results, with other values also surpassing the performance of DeYO. These findings show the robustness of our methods across a variety of hyperparameter settings.

5.4. Discussion

The Number of Samples Used. Fig. 5 compares the count of classes for each group of selected samples for the No Adapt, SAR, and SAR+FATA. With SAR, a significantly larger number of classes are selected more than 5 times, whereas without any TTA method, many classes are sampled 5 times or fewer. When our method is incorporated with SAR, considerably more classes are sampled more frequently compared to SAR alone. This increased sampling frequency contributes to the performance boost provided by FATA, as shown by Fig. 1b, which depicts the positive correlation between the sampling frequency and the accuracy.

Analysis on FATA Loss. The FATA loss demonstrates its effectiveness with the experimental results in Sec. 5.3, in contrast to MSE Loss and Simple CE. With MSE Loss, the model tends to collapse in order to align two distribution, as it is the trivial solution to achieve alignment under the random feature augmentation by producing the same output for any input. Simple CE improves the model performance, but less than the FATA loss, although the main difference is the hard label generated by the argmax operator. In contrast, the FATA loss mitigates the trivial solution by replacing the output from the original feature with a pseudo-label. Further research should further develop a theoretical analysis for the FATA loss.

Limitation. Although our method effectively enhances the model performance, our method does not have the capability to prevent the collapse phenomenon. For example, as shown in Tab. 3, the accuracy of SAR on fog corruption type is 2.77%, while it decreases to 1.56% with SAR+FATA. Future research should explore a method to prevent the collapse phenomenon.

6. Conclusion

In this paper, we propose a test-time adaptation method named Feature Augmented Test-Time Adaptation (FATA). This method fully utilizes the target samples through the feature augmentation technique, addressing the issue of limited samples from sample selection based entropy minimization methods. FATA can be seamlessly integrated into any method that employing entropy-based sampling, allowing a model to leverage target samples more effectively with reliably selected samples. FATA boosts the performance of existing methods across various network architectures. Extensive experiments, including challenging scenarios, validate the effectiveness and robustness of FATA.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II191126, Self-learning based Autonomic IoT Edge Computing) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00356597, NRF-2022M3J6A1063021).

References

[1] 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. 1, 2

[2] 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. 3

[3] 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 *International Conference on Learning Representations*, 2020. 1

[4] Mohammad Fahes, Tuan-Hung Vu, Andrei Bursuc, Patrick Pérez, and Raoul de Charette. Poda: Prompt-driven zero-shot domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18623–18633, 2023. 7

[5] Qi Fan, Mattia Segu, Yu-Wing Tai, Fisher Yu, Chi-Keung Tang, Bernt Schiele, and Dengxin Dai. Towards robust object detection invariant to real-world domain shifts. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 3, 4, 7

[6] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020. 2

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

[8] 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, 2

[9] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018. 1, 2, 3, 5, 6, 7

[10] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3

[11] 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. 1

[12] Jungsoo Lee, Debasmit Das, Jaegul Choo, and Sungha Choi. Towards open-set test-time adaptation utilizing the wisdom of crowds in entropy minimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16380–16389, 2023. 2

[13] Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 3, 5, 6, 7

[14] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M. Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8886–8895, October 2021. 1, 2, 3, 4, 7

[15] Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. Ttn: A domain-shift aware batch normalization in test-time adaptation. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[16] 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? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21808–21820. Curran Associates, Inc., 2021. 1

[17] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 1

[18] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 29, 2016. 1

[19] 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. 1, 2, 3, 4, 5, 6, 7

[20] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 3, 5, 6, 7

[21] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE transactions on neural networks*, 22(2):199–210, 2010. 1

[22] Sunghyun Park, Seunghan Yang, Jaegul Choo, and Sungrack Yun. Label shift adapter for test-time adaptation under covariate and label shifts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16421–16431, 2023. 2

[23] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015. 1

[24] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, et al. Generalizing across domains via cross-gradient training. *ICLR*, 2018. 3

[25] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001. 3

[26] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019. 1

[27] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020. 1

[28] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018. 1

[29] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *International Conference on Learning Representations*, 2020. 3

[30] 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. 5, 7

[31] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR, 2019. 3

[32] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 3

[33] 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. 1, 2, 5, 6, 7

[34] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7201–7211, June 2022. 1, 3, 4, 6, 7

[35] Shuai Wang, Daoan Zhang, Zipei Yan, Jianguo Zhang, and Rui Li. Feature alignment and uniformity for test time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20050–20060, 2023. 2

[36] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems*, 35:38629–38642, 2022. 2, 6, 7

[37] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13025–13032, Apr. 2020. 3

[38] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021. 3