Neuro-Modulated Hebbian Learning for Fully Test-Time Adaptation

Yushun Tang[1,2], Ce Zhang[1], Heng Xu[1], Shuoshuo Chen[1],
Jie Cheng[2], Luziwei Leng[2], Qinghai Guo[2*], Zhihai He[1,3*]
[1]Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China
[2]Advanced Computing and Storage Laboratory, Huawei Technologies Co., Ltd., Shenzhen, China
[3]Pengcheng Laboratory, Shenzhen, China
{tangys2022, zhangc2019, xuh2022, chenss2021}@mail.sustech.edu.cn
{chengjie8, lengluziwei, guoqinghai}@huawei.com, hezh@sustech.edu.cn

Abstract

Fully test-time adaptation aims to adapt the network model based on sequential analysis of input samples during the inference stage to address the cross-domain performance degradation problem of deep neural networks. We take inspiration from the biological plausibility learning where the neuron responses are tuned based on a local synapse-change procedure and activated by competitive lateral inhibition rules. Based on these feed-forward learning rules, we design a soft Hebbian learning process which provides an unsupervised and effective mechanism for online adaptation. We observe that the performance of this feed-forward Hebbian learning for fully test-time adaptation can be significantly improved by incorporating a feedback neuro-modulation layer. It is able to fine-tune the neuron responses based on the external feedback generated by the error back-propagation from the top inference layers. This leads to our proposed neuro-modulated Hebbian learning (NHL) method for fully test-time adaptation. With the unsupervised feed-forward soft Hebbian learning being combined with a learned neuro-modulator to capture feedback from external responses, the source model can be effectively adapted during the testing process. Experimental results on benchmark datasets demonstrate that our proposed method can significantly improve the adaptation performance of network models and outperforms existing state-of-the-art methods.

1. Introduction

Although deep neural networks have achieved great success in various machine learning tasks, their performance tends to degrade significantly when there is data shift [27, 55] between the training data in the source domain and the testing data in the target domain [40]. To address the performance degradation problem, unsupervised domain adaptation (UDA) [16, 38, 50] has been proposed to fine-tune the model parameters with a large amount of unlabeled testing data in an unsupervised manner. Source-free UDA methods [33, 35, 67] aim to adapt the network model without the need to access the source-domain samples.

There are two major categories of source-free UDA methods. The first category needs to access the whole test dataset on the target domain to achieve their adaptation performance [35, 67]. Notice that, in many practical scenarios when we deploy the network model on client devices, the network model does not have access to the whole dataset in the target domain since collecting and constructing the test dataset on the client side is very costly. The second type of method, called fully test-time adaptation, only needs access to live streams of test samples [41, 64, 66], which is able to dynamically adapt the source model on the fly during the testing process. Existing methods for fully test-time adaptation mainly focus on constructing various loss functions to regulate the inference process and adapt the model based on error back-propagation. For example, the TENT method [66] updates the batch normalization module by minimizing an entropy loss. The TTT method [64] updates the feature extractor parameters according to a self-supervised loss on a proxy learning task. The TTT++ method [37] introduces a feature alignment strategy based on online moment matching.

1.1. Challenges in Fully Test-Time UDA

We recognize that most of the domain variations, such as changes in the visual scenes and image transformations or corruptions, are early layers of features in the semantic hierarchy [66]. They can be effectively captured and modeled by lower layers of the network model. From the perspective of machine learning, early representations through the lower layer play an important role to capture the pos-

* Corresponding authors.

terior distribution of the underlying explanatory factors for the observed input [1]. For instance, in deep neural network models, the early layers of the network tend to respond to corners, edges, or colors. In contrast, deeper layers respond to more class-specific features [72]. In the corruption test-time adaptation scenario, the class-specific features are always the same because the testing datasets are the corruption of the training domain. However, the early layers of models can be failed due to corruption.

Therefore, the central challenge in fully test-time UDA lies in how to learn useful early layer representations of the test samples without supervision. Motivated by this observation, we propose to explore neurobiology-inspired Hebbian learning for effective early-layer representation learning and fully test-time adaptation. It has been recognized that the learning rule of supervised end-to-end deep neural network training using back-propagation and the learning rules of the early front-end neural processing in neurobiology are unrelated [28]. The responses of neurons in biological neural networks are tuned by local pre-synaptic and post-synaptic activity, along with global variables that measure task performance, rather than the specific activity of other neurons [69].

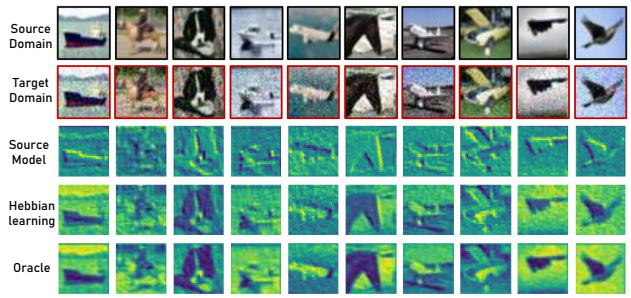


Figure 1. The feature map visualization after the first convolution layer obtained by different learning methods.

1.2. Hebbian Learning

Hebbian learning aims to learn useful early layer representations without supervision based on local synaptic plasticity rules, which is able to generate early representations that are as good as those learned by end-to-end supervised training with back-propagation [28, 52]. Drastically different from the current error back-propagation methods which require pseudo-labels or loss functions from the top network layers, Hebbian learning is a pure feed-forward adaptation process and does not require feedback from the distant top network layers. The responses of neurons are tuned based on a local synapse-change procedure and activated by competitive lateral inhibition rules [28]. During the learning process, the strength of synapses undergoes local changes that are proportional to the activity of the pre-synaptic cell and dependent on the activity of the post-synaptic cell. It also introduces local lateral inhibition between neurons within a layer, where the synapses of hidden units with strong responses are pushed toward the patterns that drive them, while those with weaker responses are pushed away from these patterns.

Existing literature has shown that early representations learned by Hebbian learning are as well as back-propagation and even more robust in testing [28, 29, 52]. Figure 1 shows the feature maps learned by different methods. The first row shows the original image. The second row shows the images in the target domain with significant image corruption. The third row shows the feature maps learned by the network model trained in the source domain for these target-domain images. The fourth row shows the feature maps learned by our Hebbian learning method. The last row ("oracle") shows the feature maps learned with true labels. We can see that the unsupervised Hebbian learning is able to generate feature maps which are as good as those from supervised learning.

1.3. Our Major Idea

In this work, we observe that Hebbian learning, although provides a new and effective approach for unsupervised learning of early layer representation of the image, when directly applied to the network model, is not able to achieve satisfactory performance in fully test-time adaptation. First, the original hard decision for competitive learning is not suitable for fully test-time adaptation. Second, the Hebbian learning does not have an effective mechanism to consider external feedback, especially the feedback from the top network layers. We observe that, biologically, the visual processing is realized through hierarchical models considering a bottom-up early representation learning for the sensory input, and a top-down feedback mechanism based on predictive coding [15, 56].

Motivated by this, in this work, we propose to develop a new approach, called *neuro-modulated Hebbian learning (NHL)*, for fully test-time adaptation. We first incorporate a soft decision rule into the feed-forward Hebbian learning to improve its competitive learning. Second, we learn a neuro-modulator to capture feedback from external responses, which controls which type of feature is consolidated and further processed to minimize the predictive error. During inference, the source model is adapted by the proposed NHL rule for each mini-batch of testing samples during the inference process. Experimental results on benchmark datasets demonstrate that our proposed method can significantly improve the adaptation performance of network models and outperforms existing state-of-the-art methods.

1.4. Summary of Major Contributions

To summarize, our major contributions include: (1) we identify that the major challenge in fully test-time adaptation lies in effective unsupervised learning of early layer representations, and explore neurobiology-inspired soft Hebbian learning for effective early layer representation learning and fully test-time adaptation. (2) We develop a new neuro-modulated Hebbian learning method which combines unsupervised feed-forward Hebbian learning of early layer representation with a learned neuro-modulator to capture feedback from external responses. We analyze the optimal property of the proposed NHL algorithm based on free-energy principles [14, 15]. (3) We evaluate our proposed NHL method on benchmark datasets for fully test-time adaptation, demonstrating its significant performance improvement over existing methods.

2. Related Work

In this section, we review existing methods related to our work including test-time adaptation, source-free UDA, domain generalization, and unsupervised Hebbian learning.

2.1. Test-time Adaptation

Test-time adaptation aims to online adapt the trained model while testing the input samples in the target domain. Sun *et al*. [64] proposed Test-time Training (TTT) by optimizing a self-supervised loss through a proxy task on the source before adapting to the target domain. The method proposed in TTT++ [37] aims to minimize the distribution shift between the training and testing feature distributions by dynamically matching the moments online. It should be noted that this method requires specific training on the source data, which is not available in fully test-time adaptation scenarios [66]. TENT [66] fine-tuned the scale and bias parameters of the batch normalization layers using an entropy minimization loss during the inference process. DUA [41] adapted the statistics of the batch normalization layer only on a tiny fraction of test data and augmented a small batch of target data to adapt the model. Choi *et al*. [8] proposed a shift-agnostic weight regularization and an auxiliary task for the alignment between the source and target features. Note that this method requires the source data for computing the source prototypes. The continual test-time adaptation methods [39, 68] consider online TTA where target data is continually changing during inference. Instead of using parameters of the pre-trained model, Boudiaf *et al*. [4] only adapted the model's output by optimizing an objective function based on Laplacian adjusted maximum-likelihood estimation. Besides image classification, test-time adaptation has been successfully applied in various machine learning tasks, such as scene deblurring [7], super-resolution [62], human pose estimation [34], image segmentation [21], object detection [41], *etc*.

2.2. Source-free Unsupervised Domain Adaptation

Recently, source-free UDA has emerged as an important research topic. It aims to adapt the source models to unlabeled target domains without accessing the data from the source domain. Among them, SHOT [35] proposed to learn target-specific features based on an information maximization criteria and pseudo-label prediction. 3C-GAN [33] generated target-style images by training a collaborative class conditional GAN module and using a clustering-based regularization scheme. G-SFDA [70] proposed a domain-specific attention mechanism that selectively activates different feature channels for different domains. CPGA [54] proposed to generate avatar prototypes instead of images via contrastive learning. HCL [22] proposed a solution for addressing the lack of source data by introducing both instance-level and category-level historical contrastive learning. DIPE [67] focuses on exploring the domain-invariant parameters of the model, rather than trying to learn domain-invariant representations. SFDA-DE [10] aligned domains by estimating source class-conditioned feature distribution and minimizing a contrastive adaptation loss function. Existing source-free methods are offline. They need to analyze the whole test dataset and update the model for a number of adaptation epochs.

2.3. Domain Generalization

Domain generalization aims to train a model only on the source data that are generalizable to unseen target domains [74]. Domain generalization methods typically aim to learn domain-invariant representations by aligning the distributions of the source domains [43, 44, 73]. Methods based on data augmentation [53, 65] for regularizing the training process have been studied. Ensemble learning methods [11, 61] generate an ensemble of different model weights from different partitions of the training data. Recently, inference-time optimization without training the network model has been studied for domain generalization. Pandey *et al*. [49] used a generative model to project the target data onto the source-feature manifold with labels being preserved by solving an optimization problem during the inference stage. Iwasawa *et al*. [25] proposed a back-propagation-free generalization method by computing distance to a pseudo-prototype representation.

2.4. Unsupervised Hebbian Learning

In traditional end-to-end training, back-propagation is usually used to update the weights of deep neural networks. It has been recognized that the learning rule of supervised end-to-end deep neural network training using back-propagation is much different from the learning rules of the early front-end neural processing in neurobiology. In

addition, supervised training of deep neural networks with back-propagation requires a large amount of labeled samples [30]. The tuning of neuron responses in biological neural networks is achieved through a physically local synapse-change procedure. The superficial cerebral cortex exhibits a common connectivity pattern in which neurons are activated through competition via lateral inhibition [2, 12]. This competition leads to the suppression of weakly activated neurons and the amplification of strongly activated ones, a phenomenon known as competitive learning or "winner-takes-all (WTA)" learning [57, 59]. Motivated by Hebb's idea [19], several biological plausibility learning rules have been proposed, where changes of the synapse strength depend only on the activities of the pre-synaptic and post-synaptic neurons. The Oja's rule [47] proposed a linear neuron model with constrained Hebbian synaptic modification and derived a new unconstrained learning method. Krotov *et al*. [28] proposed a family of biologically plausible learning rules that enable the learning of early representations that are comparable to those achieved by end-to-end supervised training with back-propagation. Pogodin *et al*. [52] presents a family of learning rules which use pre- and post-synaptic firing rates and a global teaching signal. They perform almost the same as the back-propagation method.

3. Method

In this section, we present our method of neuro-modulated Hebbian learning for fully test-time adaptation.

3.1. Problem Formulation

Suppose that a model $q_{\theta_s}(y|X_s)$ with parameters θ_s has been successfully trained on the source data $\{X_s\}$ with labels $\{Y_s\}$ where the true distribution is $p_s(y|X_s)$. It consists of a feature extractor $\mathbf{F}$ and a classifier $\mathbf{C}$. During fully test-time adaptation, we are given the target data $\{X_t\}$ with unknown labels $\{Y_t\}$. Our goal is to adapt the trained model q_{θ_s} in an unsupervised manner during testing to approximate the true target distribution $p_t(y|X_t)$. Given a sequence of input sample batches $\{B_1, B_2, ..., B_n\}$, the i-th adaptation of the network model can only rely on the i-th batch of test samples B_i.

We consider this transfer learning on a new target domain as an active inference process. The pre-trained model $q_{\theta_s}(y|X_s)$ is considered as a prior. Thus, predicting the labels for target samples becomes a Bayesian posterior $q_{\theta_1}(y|B_1)$ for the first batch, given the prior model $q_{\theta_s}(y|X_s)$. After the first adaptation batch, the prior model is updated to $q_{\theta_1}(y|B_1)$. This Bayesian inference process is repeated for all subsequent batches and produces a final posterior estimation

$$q_{\theta,t}(y|X_t) = q_{\theta_n}(y|B_n), \tag{1}$$

given the prior model $q_{\theta_{n-1}}(y|B_{n-1})$.

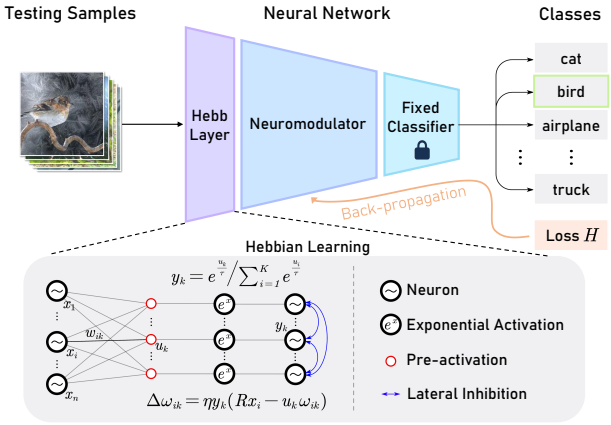


Figure 2. An overview of the proposed NHL method. During inference, the first convolution layer of the source model is fine-tuned by the Hebbian learning rule and the neuromodulator is further fine-tuned by the entropy loss before making a prediction given each mini-batch testing sample. The lock symbol means the classifier is fixed in the test-time adaptation process.

3.2. Overview of Our NHL Method

As illustrated in Figure 2, our proposed neuro-modulated Hebbian learning consists of two major components: the feed-forward soft Hebbian learning layer and the neuro-modulator. The soft Hebbian learning layer aims to learn useful early layer representations without supervision based on local synaptic plasticity and soft competitive learning rules. It is able to generate early representations which are as good as those learned by end-to-end supervised training with labeled samples and back-propagation. During our experiments, we find that this soft Hebbian learning layer can significantly improve the performance of the network model in the target domain. However, we recognize that the feed-forward Hebbian learning only is not able to achieve competitive performance as the current state-of-the-art methods for fully test-time adaptation. We find that it lacks the capability to effectively respond to external stimulus, specifically the feedback from the network output layer.

To address this issue, we propose to design a neuro-modulator layer, an intermediate layer or an interface between the soft Hebbian learning layer and the classifier module of the network. This neuro-modulator layer is updated using back-propagation with the entropy loss being evaluated at the network output. It serves as the bridge between two different learning approaches: the feed-forward Hebbian learning and the original error back-propagation. From the ablation studies summarized in Table 5, we can see that both algorithm components, the feed-forward soft Hebbian learning, and the neuro-modulator are contributing significantly to the overall performance improvement.

The proposed NHL method can be also formulated by the free-energy principle in cognitive science [14, 15, 56]. A hierarchical predictive coding model was proposed by Rao

and Ballard [56] to learn a hierarchical internal model for human perception by maximizing the posterior probability of generating the observed data. This can be realized by a concurrent process of prediction through a bottom-up feed-forward generation process (such as our Hebbian learning layer) combined with a top-down feedback-based optimization process (such as our neuro-modulator). Specifically, given the sensory input x, assume x is generated by environmental causes ϑ, denoted as $p(x) = p(x, \vartheta)$. The free-energy principle states that the brain encodes the recognition density over sensory causes [14]. Mathematically, it optimizes a generative probabilistic mixture as $q_1(x) := q(x, \vartheta)$. It has been demonstrated that this process can be achieved by a Hebbian-like learning approach [42, 45].

On the other hand, to approximate the true distribution $p_t(y|X_t)$ by a posterior approximation $q_2(y) := q_{\theta,t}(y|X_t)$, one can consider the similarity between these two distributions measured by the following Kullback-Leibler (KL) divergence

$$\mathrm{KL}[q_2(y)||p_t(y|X_t)] = \int q_2(y) \log \frac{q_2(y)}{p_t(y|X_t)} dy. \quad (2)$$

According to the analysis in Bogacz [3], this minimization of KL-divergence can be converted to minimization of the *free-energy* F defined as:

$$F = \int q_2(y) \log \frac{q_2(y)}{p_t(X_t, y)} dy. \quad (3)$$

The maximization of the free-energy F can be solved by an active inference process if we consider the adaptation for each batch of test samples as a decision-making step [13]. This leads to a feedback-based optimization, just like the neuro-modulation as proposed in our NHL method. In the following sections, we explain the two major components of our NHL method in more detail.

3.3. Feed-Forward Soft Hebbian Learning

As discussed above, the early feed-forward layer aims to learn useful layer representations of the input without any supervision. An approximate solution to this problem can be reduced to finding the first principal component of the input data [47], known as Oja's rule and its variations. The Oja's rule is itself a variation of the Hebbian rule, plus a normalization condition. In Oja's rule, the plasticity defined for a synaptic weight w_{ik} which connects a pre-synaptic neuron i with input x_i to a post-synaptic neuron k with activation y_k is:

$$\Delta w_{ik} = \eta y_k (x_i - y_k w_{ik}). \quad (4)$$

These weights are updated based on a plasticity rule [28]. We recognize that this process lacks the ability to detect different features of the data by different hidden units in the network. To address this problem, we propose to identify a good set of weights, by considering a generative model to optimize the distribution similarity to the input data.

Specifically, we assume that the target-domain input samples $\{X_t\}$ are generated by hidden causes ϑ with distribution $p_t(x)$. We define an approximation to $p_t(x)$ by $q(x|\boldsymbol{w})$ conditioned on the weights $\boldsymbol{w}$. Moreover, we use a mixture of exponential functions to define the probability:

$$q(x|\boldsymbol{w}_k) = \exp(< \boldsymbol{w}_k, x >)/N, \quad (5)$$

where $\boldsymbol{w}_k$ is the weight vector corresponds to the post-synaptic neuron k, and N is a normalization factor to ensure that q is a probabilistic measure. In the above generative model, the objective function is given by the KL divergence $\mathrm{KL}[p_t(x)|q(x|\boldsymbol{w})]$. It can be shown that the optimal parameter vector that optimizes the KL-divergence is proportional to the mean of the input distribution [45].

On the other hand, (5) corresponds to neural interpretation as the activation function with normalized weights $\boldsymbol{w}_k/R$, where R is the norm of the weight vector $\boldsymbol{w}_k$. For the network layer with K neurons, we define the output of the k-th neuron to be:

$$y_k = e^{\frac{u_k}{\tau}} / \left(\sum_{i=1}^{K} e^{\frac{u_i}{\tau}} \right), \quad (6)$$

where u_k is the k-th neuron's weighted input, *i.e.* $u_k = \sum_i w_{ik} \cdot x_i$. τ is a temperature-scaling hyper-parameter. This leads to a new soft Hebbian plasticity rule:

$$\Delta w_{ik} = \eta y_k (R x_i - u_k w_{ik}), \quad (7)$$

where η is the learning rate. It can be shown that, using the plasticity rule in (7) to update the weights, they can converge to the equilibrium which lies in the sphere of radius R. The derivative of the weight vector norm can be expressed as:

$$\begin{aligned} \frac{d \parallel \boldsymbol{w}_k \parallel}{dt} &= 2\boldsymbol{w}_k \frac{d\boldsymbol{w}_k}{dt} = 2\boldsymbol{w}_k \eta y_k (R\boldsymbol{x} - u_k \boldsymbol{w}_k) \\ &= 2\eta \boldsymbol{w}_k y_k (R\boldsymbol{x} - \boldsymbol{w}_k \boldsymbol{x} \boldsymbol{w}_k) = 2\eta u_k y_k (R - \parallel \boldsymbol{w}_k \parallel). \end{aligned} \quad (8)$$

With this, we see that the norm of the weight vector convergences to a sphere of radius R. This is because the norm of the weights decreases if $\|\boldsymbol{w}_k\| > R$ and increases if $\|\boldsymbol{w}_k\| < R$. More details are provided in the Supplemental Materials.

3.4. The Neuro-Modulation Layer

From the ablation studies in Section 4, we can see that the above soft Hebbian learning alone does not automatically lead to a perfect posterior estimation for $p_t(y|X_t)$. It cannot achieve the state-of-the-art performance for fully test-time adaptation. This is because it does not have an effective mechanism to consider external feedback, especially

the feedback from the top network layers. Our proposed solution is to incorporate one or more modulating layers to steer the updates of weights to the desired outcome [17, 36]. This so-called neuro-modulator has been explored in neuroscience research [5, 48, 51, 63]. Recent research [32] shows that the level of neuro-modulation may change the process of synaptic consolidation, thus ultimately controlling which type of information is consolidated in the upper neural network. Unlike the above neuromodulator-based learning, where the modulator factor is embedded inside the Hebbian rule, we consider such a neuromodulation process in a disentangled way derived from the free-energy principle. It serves as an interface between the feed-forward Hebbian layer and the top decision network.

As defined in (3), the problem of minimizing the KL-divergence for $q_2(y)$ and its true posterior $p_t(y|X_t)$ can be formulated based on the free-energy principle:

$$\mathrm{KL}[q_2(y)||p_t(y|X_t)] = F + \log P_t(X_t), \tag{9}$$

where $P_t(X_t) := \int q_2(y)p_t(X_t)dy = p_t(X_t)$ is the normalization term. Note that this term does not depend on $q_2(y)$. Therefore, minimizing the KL-divergence is reduced to minimizing F. To this end, given a batch B of data in the target domain, we rewrite (use $p_t(y, B) = p_t(B|y)p_t(y)$) and decompose the free-energy F_B for current batch into the following two items:

$$F_B = \mathrm{KL}[q_2(y|B)||p_t(y)] - \int_y q_2(y|B) \log p_t(B|y)dy. \tag{10}$$

The first term in (10) is already minimized through soft Hebbian learning, while minimizing the second term requires the likelihood distribution $p_t(B|y)$. Since $p_t(B|y) = p_t(y|B)p_t(B)/p_t(y)$ and $q_2(y|B)$ is considered as an approximation of $p_t(y|B)$, we minimize the entropy of y given the current batch B in a discrete way as:

$$\arg\min_{\boldsymbol{w}} \; H(y|B) = \arg\max_{\boldsymbol{w}} \sum q_2(y|B) \log q_2(y|B), \tag{11}$$

where $\boldsymbol{w}$ are the weights in the layer implementing the neuro-modulator. As in existing deep neural network training, we can use gradient descent to optimize these weights.

4. Experiments

In this section, we conduct experiments on multiple test-time adaptation benchmark datasets to evaluate the performance of our proposed NHL method.

4.1. Benchmark Datasets

We evaluate our method among the following popular benchmark datasets for test-time adaptation. (1) **CIFAR-10/100C.** We choose CIFAR-10/100 [26] with 10/100 classes, a source training set of $50,000$, and a target testing set CIFAR-10/100C [20] of $10,000$ for small-size image experiments at an accessible scale. (2) **ImageNet-C.** We choose the ImageNet [58] with $1,000$ classes, a source training set of 1.2 million, and a validation set of $50,000$ for large-size image experiments. It should be noted that we use a fixed target testing subset of the validation set with 5000 images on which all models are evaluated following the RobustBench protocol [9]. (3) **SVHN→MNIST/MNIST-M/USPS.** Following TENT for domain adaptation, we choose a SVHN [46] source model and transfer it to MNIST [31] ($10,000$ test samples), MNIST-M [16] ($10,000$ test samples) and USPS [23] ($2,007$ test samples), respectively.

4.2. Comparison Methods

We compare our method against the following fully test-time adaptation methods: (1) **Source:** the baseline model is trained only on the source data without any fine-tuning during the test process. (2) **TTT** [64]: it adapts the feature extractor by optimizing a self-supervised loss through a proxy task. However, it requires training the same proxy task on the source domain. (3) **NORM** [60]: this test-time normalization method updates the batch normalization [24] statistics using the mini-batch samples during the test process. (4) **TENT** [66]: it fine-tunes scale and bias parameters of the batch normalization layers using an entropy minimization loss during inference. (5) **DUA** [41]: it adapts the statistics of the batch normalization layer only on a tiny fraction of test data and augments a small batch of target data to adapt the model.

4.3. Implementation Details

Following the official implementations of TTT[1], TENT[2] and DUA[3], we use the ResNet-26 [18], Wide-ResNet-28-10 [71], and Wide-ResNet-40-2 [71] as the backbone networks for the CIFAR-10C dataset. We use the Wide-ResNet-40-2 network for the CIFAR-100C dataset. For the ImageNet-C dataset, we use the ResNet-18 [18] backbone network. We use the pre-trained model weights from the official implementations of TENT or DUA for all backbones based on the RobustBench protocol [9]. For the digit recognition transfer tasks, we use the pre-trained model weights of SVHN from the *pytorch-playground* [6]. For fair performance comparisons, all methods in each experimental condition share the same architecture and the same pre-trained model parameters. The batch size is set to 128. More implementation details are provided in the Supplemental Materials.

[1] TTT: https://github.com/yueatsprograms/ttt_cifar_release
[2] TENT: https://github.com/DequanWang/tent
[3] DUA: https://github.com/jmiemirza/DUA

Table 1. Top-1 Classification Error (%) for each corruption in **CIFAR-10C** at the highest severity (Level 5). For TTT, TENT, and DUA, we use the **ResNet-26** (top), **WRN-28-10** (middle), and **WRN-40-2** (bottom) from their official implementation. The smallest error is shown in **bold**.

Methods	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	Avg.
Source	67.7	63.1	69.9	55.3	56.6	42.2	50.1	31.6	46.3	39.1	17.1	74.6	34.2	57.9	31.7	49.2
TTT [64]	45.6	41.8	50.0	21.8	46.1	23.0	23.9	29.9	30.0	25.1	**12.2**	23.9	**22.6**	47.2	**27.2**	31.4
NORM [60]	44.6	43.7	49.1	29.4	45.2	26.2	26.9	25.8	27.9	23.8	18.3	34.3	29.3	37.0	32.5	32.9
TENT [66]	39.4	38.8	47.9	19.9	45.0	23.2	20.6	28.1	32.1	24.5	16.1	26.7	32.4	30.6	35.5	30.7
DUA [41]	34.9	32.6	42.2	18.7	**40.2**	24.0	18.4	**23.9**	**24.0**	20.9	12.3	27.1	27.2	26.2	28.7	26.8
Ours	**33.2**	**30.6**	**38.2**	**17.7**	41.2	**20.8**	**17.4**	24.0	27.2	**20.4**	13.5	**21.1**	28.4	**23.7**	28.9	**25.8**
Source	72.3	65.7	72.9	46.9	54.3	34.8	42.0	25.1	41.3	26.0	9.3	46.7	26.6	58.5	30.3	43.5
NORM [60]	28.1	26.1	36.3	12.8	35.3	14.2	12.1	17.3	17.4	15.3	8.4	12.6	23.8	19.7	27.3	20.4
TENT [66]	24.8	23.5	33.0	12.0	31.8	13.7	**10.8**	15.9	16.2	13.7	7.9	12.1	22.0	17.3	24.2	18.6
DUA [41]	27.4	24.6	35.3	13.1	34.9	14.6	11.6	16.8	17.5	13.1	**7.6**	14.1	22.7	19.3	26.2	19.9
Ours	**23.6**	**21.4**	**30.9**	**11.0**	**31.1**	**13.0**	10.9	**14.2**	**15.5**	**13.0**	8.0	**10.3**	**21.8**	**16.7**	**22.4**	**17.6**
Source	28.8	22.9	26.2	9.5	20.6	10.6	9.3	14.2	15.3	17.5	7.6	20.9	14.7	41.3	14.7	18.3
NORM [60]	18.7	16.4	22.3	9.1	22.1	10.5	9.7	13.0	13.2	15.4	7.8	12.0	16.4	15.1	17.6	14.6
TENT [66]	15.7	13.2	18.8	7.9	18.1	9.0	8.0	10.4	10.8	12.4	6.7	10.0	14.0	11.4	14.8	12.1
DUA [41]	15.4	13.4	17.3	8.0	18.0	9.1	**7.7**	10.8	10.8	12.1	6.6	10.9	13.6	13.0	14.3	12.1
Ours	**13.4**	**12.3**	**15.0**	**7.5**	**16.0**	**8.7**	**7.7**	**9.1**	**9.6**	**10.1**	**6.4**	**8.2**	**13.3**	**9.3**	**13.3**	**10.7**

Table 2. Top-1 Classification Error (%) for each corruption in **CIFAR-100C** at the highest severity (Level 5).

Methods	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	Avg.
Source	65.7	60.1	59.1	32.0	51.0	33.6	32.4	41.4	45.2	51.4	31.6	55.5	40.3	59.7	42.4	46.7
NORM [60]	44.7	44.2	47.4	32.4	46.4	32.9	33.0	39.0	38.4	45.3	30.2	36.6	40.6	37.2	44.2	39.5
TENT [66]	40.3	39.9	41.8	29.8	42.3	31.0	30.0	34.5	35.2	39.5	28.0	33.9	38.4	33.4	41.4	36.0
DUA [41]	42.2	40.9	41.0	30.5	44.8	32.2	29.9	38.9	37.2	43.6	29.5	39.2	39.0	35.3	41.2	37.6
Ours	**38.4**	**37.1**	**36.2**	**28.4**	**41.0**	**29.3**	**29.7**	**32.2**	**33.1**	**36.1**	**26.4**	**30.9**	**36.2**	**30.8**	**38.3**	**33.6**

Table 3. Top-1 Classification Error (%) for each corruption in **ImageNet-C** at the highest severity (Level 5).

Methods	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	Avg.
Source	98.9	97.6	99.2	93.3	89.0	90.2	82.3	87.9	92.0	99.5	75.9	99.5	65.3	60.3	54.0	85.7
TTT [64]	75.5	76.8	81.9	89.6	82.8	79.1	71.3	83.6	81.0	98.3	59.0	99.0	54.7	53.2	**49.6**	75.7
NORM [60]	60.2	60.7	59.8	76.6	68.7	67.4	64.2	64.6	66.2	74.7	57.0	88.8	55.8	53.0	52.3	64.7
TENT [66]	59.4	59.6	58.7	**72.5**	66.1	64.9	62.1	62.2	64.9	68.6	55.2	97.9	54.5	52.1	51.7	62.7
DUA [41]	71.9	72.6	72.4	90.2	80.8	83.1	74.7	76.4	77.9	87.3	62.6	99.3	60.8	58.4	52.6	74.7
Ours	**56.7**	**56.7**	**56.6**	73.3	**65.7**	**61.0**	**62.0**	**58.6**	**63.3**	**63.9**	**53.1**	**77.5**	**54.0**	**52.0**	51.5	**60.4**

4.4. Performance Results

The classification errors in the highest severity level corruption test datasets for test-time adaptation are reported in Tables 1, 2 and 3, with results of comparison methods directly cited from their original papers. Table 1 compares the classification error of our proposed method against recent test-time adaptation methods on the CIFAR-10C dataset. Our method performs better than other baselines with the three backbones including ResNet-26, WRN-28-10, and WRN-40-2, indicating the effectiveness of the proposed test-time adaptation method. Table 2 shows the performance comparison results on the CIFAR-100C dataset. Very encouraging results are also obtained on the large-size complicated ImageNet-C dataset, as shown in Table 3. The mean adaptation error of the full test dataset on Gaussian noise of CIFAR-10C in the different adaptation stages is shown in Figure 3. We can see that our method outperforms the NORM and TENT methods after the 5-th batch test-time adaptation. This implies our method can reduce error faster given few testing samples. Results for lower severity levels of corruption are provided in Supplemental Materials.

Results for digit recognition from the SVHN to MNIST, MNIST-M, and USPS datasets are also reported in Table 4. All experiments use the same open-source source model. Note that the result for the TENT method is reproduced here since the pre-trained model for the TENT method was not provided and explained in its original paper. We can see that our method achieves the lowest average error when

compared to other test-time adaptation methods. The performance improvement is quite impressive.

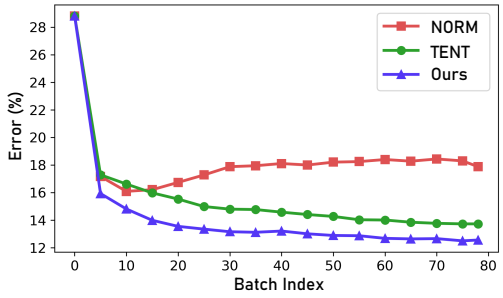


Figure 3. The mean adaptation error of the full test dataset on Gaussian noise of CIFAR-10C in the different adaptation stages.

Table 4. Top-1 Classification Error (%) for test-time adaptation on digit recognition. The asterisk (*) means the implementation with a *pytorch-playground* [6] pre-trained source model by us.

Methods	MNIST	MNIST-M	USPS	Avg.
NORM* [60]	39.6	52.1	41.4	44.4
TENT* [66]	45.8	56.2	48.3	50.1
Ours	**31.2**	**47.9**	**32.6**	**37.2**

4.5. Further Performance Analysis

4.5.1 Ablation Study

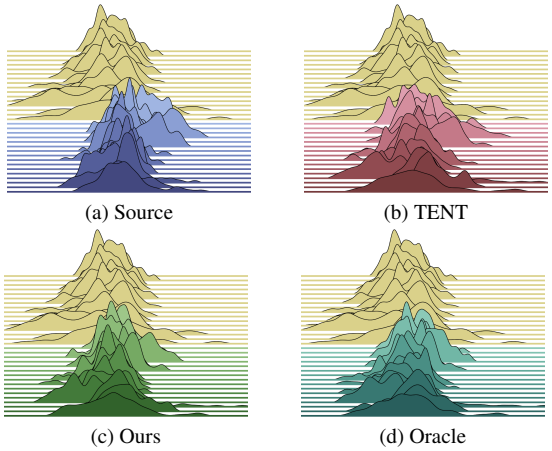
We conduct the ablation study with test-time adaptation tasks on the ImageNet-C dataset to investigate the contribution of our method. Hebbian learning alone does not automatically lead to a perfect posterior estimation for $p_t(y|X_t)$ due to the lack of global information communication. From Table 5, we can see that feed-forward soft Hebbian Learning plays a significant effect and fine-tuning the feedback neuro-modulator of Block 1/2 also improves the performance. The average error is increased when expanding the neuro-modulator to more blocks. It is because optimizing more parameters with a mini-batch of testing samples is becoming more challenging.

Table 5. Ablation study on **ImageNet-C** based on ResNet-18 including 4 Blocks at the highest severity (Level 5).

Methods	Avg. Error
Source (without adaptation)	85.7
BP Conv1	85.0
Hebbian Conv1	67.2
Hebbian Conv1 + Neuromodulator (Block 1)	60.5
Hebbian Conv1 + Neuromodulator (Block 1/2)	**60.4**
Hebbian Conv1 + Neuromodulator (Block 1/2/3)	61.2
Hebbian Conv1 + Neuromodulator (Block 1/2/3/4)	64.8

4.5.2 Feature Visualization

Figure 4 compares the feature distributions on corrupted data obtained by different adaptation methods, including the source model with no adaptation, the TENT method, and our method. We also include the feature distribution for the supervised learning with the labeled sample, denoted by "Oracle". We can see that our method is able to learn features that are close to those obtained by supervised learning.



(a) Source (b) TENT (c) Ours (d) Oracle

Figure 4. Density plots of test-time adapted features distribution on CIFAR-10-C with Gaussian noise (front) and reference features without corruption (back with yellow color). Each horizontal axis corresponds to one channel. The height of each ridge indicates the number of features that take the same value.

5. Further Discussions

Hebbian Learning by competition through lateral inhibition is a feed-forward process that has no gradient. If combining Hebbian Learning with back-propagation, it is limited to propagate the gradient through Hebbian layers to earlier gradient-based layers during training. Therefore, the Hebbian layers can only be designed in the early layers of models without back-propagation and gradient. In addition, although the Hebbian learning rule is commonly used for long-term reinforcement, the Hebb principle does not cover all forms of long-term synaptic plasticity.

6. Conclusion

In this work, we take inspiration from the biological neuromodulation process to construct a novel neuro-modulated Hebbian Learning (NHL) framework. With the unsupervised feed-forward soft Hebbian learning being combined with a learned neuro-modulator to capture feedback from external responses, the source model can be effectively adapted during the testing process. Experimental results on benchmark datasets demonstrate that our proposed method can significantly reduce the testing error for image classification with corruption, and reach new state-of-the-art performance.

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE TPAMI*, 35(8):1798–1828, 2013. 2

[2] Tom Binzegger, Rodney J Douglas, and Kevan AC Martin. A quantitative map of the circuit of cat primary visual cortex. *Journal of Neuroscience*, 24(39):8441–8453, 2004. 4

[3] Rafal Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of Mathematical Psychology*, 76:198–211, 2017. 5

[4] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *CVPR*, pages 8344–8353, 2022. 3

[5] K Bougrova, N Bonacchi, JA Catarino, EE Dewitt, GT Meijer, P Dayan, ZF Mainen, et al. Comparison of neuromodulator signaling in a visual decision-making task. In *Annual Meeting of the Society for Neuroscience*, 2022. 6

[6] Aaron Chen and Gus Smith. pytorch-playground. https://github.com/aaron-xichen/pytorch-playground, 2020. 6, 8

[7] Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *CVPR*, pages 9137–9146, 2021. 3

[8] Sungha Choi, Seunghan Yang, Seokeon Choi, and Sungrack Yun. Improving test-time adaptation via shift-agnostic weight regularization and nearest source prototypes. *arXiv preprint arXiv:2207.11707*, 2022. 3

[9] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *NeurIPS*, 2021. 6

[10] Ning Ding, Yixing Xu, Yehui Tang, Chao Xu, Yunhe Wang, and Dacheng Tao. Source-free domain adaptation via distribution estimation. In *CVPR*, pages 7212–7222, 2022. 3

[11] Zhengming Ding and Yun Fu. Deep domain generalization with structured low-rank constraint. *IEEE TIP*, 27(1):304–313, 2017. 3

[12] Rodney J Douglas, Kevan AC Martin, et al. Neuronal circuits of the neocortex. *Annual Review of Neuroscience*, 27(1):419–451, 2004. 4

[13] Karl Friston. Hierarchical models in the brain. *PLoS Computational Biology*, 4(11):e1000211, 2008. 5

[14] Karl Friston. The free-energy principle: a rough guide to the brain? *Trends in Cognitive Sciences*, 13(7):293–301, 2009. 3, 4, 5

[15] Karl Friston, James Kilner, and Lee Harrison. A free energy principle for the brain. *Journal of Physiology, Paris*, 100(1-3):70–87, 2006. 2, 3, 4

[16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189. PMLR, 2015. 1, 6

[17] Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Frontiers in Neural Circuits*, 12:53, 2018. 6

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[19] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005. 4

[20] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2018. 6

[21] Minhao Hu, Tao Song, Yujun Gu, Xiangde Luo, Jieneng Chen, Yinan Chen, Ya Zhang, and Shaoting Zhang. Fully test-time adaptation for image segmentation. In *MICCAI*, pages 251–260. Springer, 2021. 3

[22] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In *NeurIPS*, volume 34, pages 3635–3649, 2021. 3

[23] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE TPAMI*, 16(5):550–554, 1994. 6

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456. PMLR, 2015. 6

[25] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *NeurIPS*, volume 34, pages 2427–2440, 2021. 3

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009. 6

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[28] Dmitry Krotov and John J Hopfield. Unsupervised learning by competing hidden units. *Proceedings of the National Academy of Sciences*, 116(16):7723–7731, 2019. 2, 4, 5

[29] Gabriele Lagani, Fabrizio Falchi, Claudio Gennaro, and Giuseppe Amato. Hebbian semi-supervised learning in a sample efficiency setting. *Neural Networks*, 143:719–731, 2021. 2

[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 4

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[32] Andrew B Lehr, Jannik Luboeinski, and Christian Tetzlaff. Neuromodulator-dependent synaptic tagging and capture retroactively controls neural coding in spiking neural networks. *Scientific Reports*, 12(1):1–18, 2022. 6

[33] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, pages 9641–9650, 2020. 1, 3

[34] Yizhuo Li, Miao Hao, Zonglin Di, Nitesh Bharadwaj Gundavarapu, and Xiaolong Wang. Test-time personalization with a transformer for human pose estimation. In *NeurIPS*, volume 34, pages 2583–2597, 2021. 3

[35] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, pages 6028–6039. PMLR, 2020. 1, 3

[36] Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020. 6

[37] 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 *NeurIPS*, volume 34, pages 21808–21820, 2021. 1, 3

[38] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105. PMLR, 2015. 1

[39] Robert A Marsden, Mario Döbler, and Bin Yang. Gradual test-time adaptation by self-training and style transfer. *arXiv preprint arXiv:2208.07736*, 2022. 3

[40] Muhammad Jehanzeb Mirza, Cornelius Buerkle, Julio Jarquin, Michael Opitz, Fabian Oboril, Kay-Ulrich Scholl, and Horst Bischof. Robustness of object detectors in degrading weather conditions. In *ITSC*, pages 2719–2724. IEEE, 2021. 1

[41] M Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *CVPR*, pages 14765–14775, 2022. 1, 3, 6, 7

[42] Timoleon Moraitis, Dmitry Toichkin, Adrien Journé, Yansong Chua, and Qinghai Guo. Softhebb: Bayesian inference in unsupervised hebbian soft winner-take-all networks. *Neuromorphic Computing and Engineering*, 2(4):044017, 2022. 5

[43] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, pages 5715–5725, 2017. 3

[44] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, pages 10–18. PMLR, 2013. 3

[45] Bernhard Nessler, Michael Pfeiffer, Lars Buesing, and Wolfgang Maass. Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Computational Biology*, 9(4):e1003037, 2013. 5

[46] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 6

[47] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982. 4, 5

[48] John O'Donnell, Douglas Zeppenfeld, Evan McConnell, Salvador Pena, and Maiken Nedergaard. Norepinephrine: a neuromodulator that boosts the function of multiple cell types to optimize cns performance. *Neurochemical Research*, 37(11):2496–2512, 2012. 6

[49] Prashant Pandey, Mrigank Raman, Sumanth Varambally, and Prathosh Ap. Generalization on unseen domains via inference-time label-preserving target projections. In *CVPR*, pages 12924–12933, 2021. 3

[50] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, volume 32, 2018. 1

[51] Marina R Picciotto, Michael J Higley, and Yann S Mineur. Acetylcholine as a neuromodulator: cholinergic signaling shapes nervous system function and behavior. *Neuron*, 76(1):116–129, 2012. 6

[52] Roman Pogodin and Peter Latham. Kernelized information bottleneck leads to biologically plausible 3-factor hebbian learning in deep networks. In *NeurIPS*, volume 33, pages 7296–7307, 2020. 2, 4

[53] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *CVPR*, pages 12556–12565, 2020. 3

[54] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. In *IJCAI*, pages 2921–2927, 2021. 3

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

[56] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999. 2, 4, 5

[57] David E Rumelhart and David Zipser. Feature discovery by competitive learning. *Cognitive Science*, 9(1):75–112, 1985. 4

[58] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 6

[59] Ueli Rutishauser, Rodney J Douglas, and Jean-Jacques Slotine. Collective stability of networks of winner-take-all circuits. *Neural Computation*, 23(3):735–773, 2011. 4

[60] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, volume 33, pages 11539–11551, 2020. 6, 7, 8

[61] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *ECCV*, pages 68–83. Springer, 2020. 3

[62] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *CVPR*, pages 3118–3126, 2018. 3

[63] Luisa Speranza, Umberto di Porzio, Davide Viggiano, Antonio de Donato, and Floriana Volpicelli. Dopamine: The neuromodulator of long-term synaptic plasticity, reward and movement control. *Cells*, 10(4):735, 2021. 6

[64] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, volume 119, pages 9229–9248. PMLR, 2020. 1, 3, 6, 7

[65] Riccardo Volpi and Vittorio Murino. Addressing model vulnerability to distributional shifts over image transformation sets. In *ICCV*, pages 7980–7989, 2019. 3

[66] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2020. 1, 3, 6, 7, 8

[67] Fan Wang, Zhongyi Han, Yongshun Gong, and Yilong Yin. Exploring domain-invariant parameters for source free domain adaptation. In *CVPR*, pages 7151–7160, June 2022. 1, 3

[68] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, pages 7201–7211, 2022. 3

[69] James CR Whittington and Rafal Bogacz. Theories of error back-propagation in the brain. *Trends in cognitive sciences*, 23(3):235–250, 2019. 2

[70] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *ICCV*, pages 8978–8987, October 2021. 3

[71] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*. British Machine Vision Association, 2016. 6

[72] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014. 2

[73] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. In *NeurIPS*, volume 33, pages 16096–16107, 2020. 3

[74] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE TPAMI*, 2022. 3