Test-Time Domain Adaptation by Learning Domain-Aware Batch Normalization

Yanan Wu[1,2*], **Zhixiang Chi**[3*], **Yang Wang**[4], **Konstantinos N. Plataniotis**[3], **Songhe Feng**[1,2†]

[1]Key Laboratory of Big Data & Artificial Intelligence in Transportation,
Ministry of Education, Beijing Jiaotong University, Beijing, 100044, China
[2]School of Computer and Information Technology, Beijing Jiaotong University, Beijing, 100044, China
[3]The Edward S Rogers Sr. ECE Department, University of Toronto, Toronto, M5S3G8, Canada
[4]Department of Computer Science and Software Engineering, Concordia University, Montreal, H3G2J1, Canada
{ynwu0510,shfeng}@bjtu.edu.cn, zhixiang.chi@mail.utoronto.ca, yang.wang@concordia.ca, kostas@ece.utoronto.ca

Abstract

Test-time domain adaptation aims to adapt the model trained on source domains to unseen target domains using a few unlabeled images. Emerging research has shown that the label and domain information is separately embedded in the weight matrix and batch normalization (BN) layer. Previous works normally update the whole network naively without explicitly decoupling the knowledge between label and domain. As a result, it leads to knowledge interference and defective distribution adaptation. In this work, we propose to reduce such learning interference and elevate the domain knowledge learning by only manipulating the BN layer. However, the normalization step in BN is intrinsically unstable when the statistics are re-estimated from a few samples. We find that ambiguities can be greatly reduced when only updating the two affine parameters in BN while keeping the source domain statistics. To further enhance the domain knowledge extraction from unlabeled data, we construct an auxiliary branch with label-independent self-supervised learning (SSL) to provide supervision. Moreover, we propose a bi-level optimization based on meta-learning to enforce the alignment of two learning objectives of auxiliary and main branches. The goal is to use the auxiliary branch to adapt the domain and benefit main task for subsequent inference. Our method keeps the same computational cost at inference as the auxiliary branch can be thoroughly discarded after adaptation. Extensive experiments show that our method outperforms the prior works on five WILDS real-world domain shift datasets. Our method can also be integrated with methods with label-dependent optimization to further push the performance boundary. Our code is available at https://github.com/ynanwu/MABN.

Introduction

Deep models achieve astonishing performance due to the matched training and testing data distributions (Choi et al. 2018; Wu et al. 2021). However, such assumption is vulnerable in the real world as it is impossible to collect training data to cover the universal distribution. Therefore, unseen distributions at inference lead to degenerate performance stemming from distribution shift (Geirhos et al. 2018).
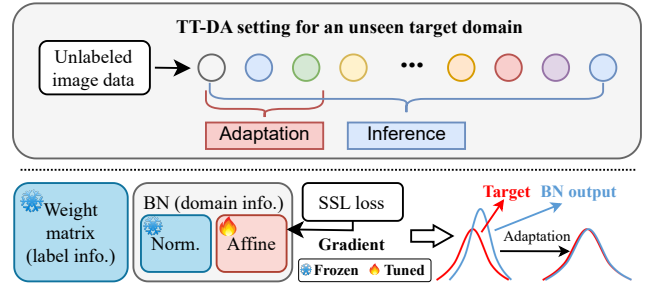

Figure 1: Top: Illustration of TT-DA setting. Given an unseen target domain at test-time, a few unlabeled data are used for adaptation and the adapted model is then used for inference. Bottom: To maximize the extraction of domain knowledge from unlabeled data, we only update the two affine parameters in BN using the label-independent self-supervised loss. The goal is to enforce the adapted affine parameters to correct the feature distribution towards target domain.

Unsupervised domain adaptation (UDA) is a line of research to mitigate the distribution shift by incorporating mutual dependence of labeled source data and unlabeled target data (Wang and Deng 2018; Ganin et al. 2016; Shu et al. 2018; Long et al. 2018). Clearly, it is impractical to repetitively perform UDA for every unseen target domain. In contrast, domain generalization (DG) aims to overcome this limitation by aspiring to train a model on source data that can effectively generalize across unseen target domains (Muandet, Balduzzi, and Schölkopf 2013; Seo et al. 2020). Nevertheless, it is unrealistic to expect a universal model to handle all diverse unseen domains. To address such limitation, pioneer works such as ARM (Zhang et al. 2021) and Meta-DMoE (Zhong et al. 2022) propose to introduce an additional phase where the model is adapted to *each* target domain using a few unlabeled data prior to inference. We refer to this scenario as test-time domain adaptation (TT-DA) as shown in the top of Fig. 1. The motivation behind TT-DA is that a small number of unlabeled data in a target domain are easy to obtain (e.g., images obtained during camera calibration) and they provide a cue of the underlying distribution of the target domain (Zhang et al. 2021).

Adapting a model with a limited amount of unlabeled

---

data remains a challenge. ARM tackles the problem by meta-learning adaptive modules for unsupervised adaptation (Finn, Abbeel, and Levine 2017). Its main limitation is that the inner and outer loop optimizations are performed on the *same* batch of unlabeled data. Intuitively, the adaptive modules are optimized to only adapt to a *single* batch instead of the broader data distribution. Unfortunately, this can impede effective generalization. Under more challenging real-world benchmarks, ARM even sometimes falls behind the baseline of Empirical Risk Minimization (ERM) as shown in the WILDS leaderboard (Koh et al. 2021). In contrast, Meta-DMoE enforces the adapted model to generalize on a disjoint query set during training to improve the overall performance. However, Meta-DMoE formulates the adaptation as a knowledge distillation process by querying the target-related knowledge from a group of teacher models trained on source domains. The adaptation is inherently bounded by the teacher models. In addition, the size of the teacher ensemble grows with the number of source domains, amplifying computational demands and considerably slowing down the adaptation process. It is worth noting that both ARM and Meta-DMoE lack a deliberate decoupling of knowledge between domain and label. This potential overlap of knowledge can introduce interference, rendering the models susceptible to performance degradation (Standley et al. 2020).

In this work, we propose a simple yet effective solution to enhance the refinement of domain knowledge acquisition, achieved by decoupling the label-related knowledge. Our work is partly inspired by the observation that the weight matrix tends to encapsulate label information, while domain-specific knowledge is embedded within the BN layer (Li et al. 2016). We propose a strategic manipulation of the BN layer to optimize the acquisition and transference of domain-specific knowledge. The BN layer normalizes the input feature followed by re-scaling and shifting using two affine parameters. However, the normalization statistics computed for the target domain under TT-DA can be unstable since we only have a small batch of examples from the target domain. Instead, we propose to only adapt the two affine parameters while directly using the normalization statistics learned from source domains during training. The intuition is that the features will first be normalized towards the source distribution, but the adapted affine parameters aim to pull the normalized feature towards the target distribution, as shown in the bottom of Fig. 1. Without the ambiguity caused by unstable statistic estimation, optimization of learning domain knowledge at the adaptation step is much more stable. On the other hand, such a strategy also keeps the same computational cost during inference without incurring additional operations (Du et al. 2020) or parameters (Bronskill et al. 2020) that are laborious to learn better statistics. Furthermore, to generate a domain-oriented supervision signal on the unlabeled data, we construct an auxiliary branch and employ class-independent self-supervised learning (SSL) (Grill et al. 2020; Liang et al. 2022).

Overall, we perform two-phase training. In the first phase, we train the whole model to learn the label knowledge and normalization statistics by mixing all the source data. To endow the affine parameters with the adaptive capabilities to new domains, in the second phase, we employ a bi-level optimization as in meta-learning. Concretely, we treat every source domain as one "task" and use few-shot unlabeled images to update the affine parameters (while keeping other model parameters fixed) via an auxiliary branch. To ensure that the optimization of the affine parameters is aligned with the main task, we define the meta-objective by evaluating the adapted affine parameters on a disjoint set in the task. Note that the affine parameters are optimized at the meta-level to act as an initialization for adaptation (Finn, Abbeel, and Levine 2017). Through such a learning paradigm, our model is learned to effectively adapt to a domain using unlabeled data and use the adapted model to perform inference. We dub our method as **Meta-Adaptive BN** (MABN). Our contributions are summarized as follows:

- We propose a simple yet effective unsupervised adaptive method that is tailored for TT-DA. We adapt only the affine parameters via a self-supervised loss to each target domain to elevate the domain knowledge learning.

- We employ a bi-level optimization to align the learning objective with the evaluation protocol to yield affine parameters that are capable of adapting domain knowledge.

- We conduct extensive experiments to show that our method is more effective in learning the domain knowledge. Thus, our domain-adapted model can be seamlessly integrated with the entropy-based TTA method (e.g. TNET (Wang et al. 2021)) where the optimization is more toward label knowledge.

- We surpass ARM and Meta-DMoE by 9.7% and 4.3% of Macro F1 on WILDS-iWildCam. We achieve superior performance on five real-world domain shift benchmarks in WILDS with both classification and regression tasks.

Related Work

Domain shift. *Unsupervised Domain Adaptation* (UDA) addresses the distribution shift by jointly training on the labeled source and unlabeled target data. Popular methods include aligning the statistical discrepancy across various distributions (Peng et al. 2019) and developing common feature space via adversarial training (Pei et al. 2018; Long et al. 2018). Recently, *source-free* UDA methods have been proposed to allow the absence of source data, such as generative-based methods (Li et al. 2020; Qiu et al. 2021). However, both settings are unrealistic. They have to access sufficiently large unlabeled target datasets, and their objective is to achieve high performance on that particular target domain. *Domain Generalization* (DG) is another line of research that learns a generic model from one or several source domains, and expects it to generalize well on unseen target domains (Zhou et al. 2020; Lv et al. 2022). Both types of domains cannot be accessed simultaneously. However, deploying such model to all unseen target domains fails to explore domain specialty and often yields inferior solutions.

Test-time adaptation/training (TTA) aims to adapt models towards test data to overcome the performance degradation caused by distribution shifts. (Sun et al. 2020) employ rotation prediction to update the model during inference. (Chi
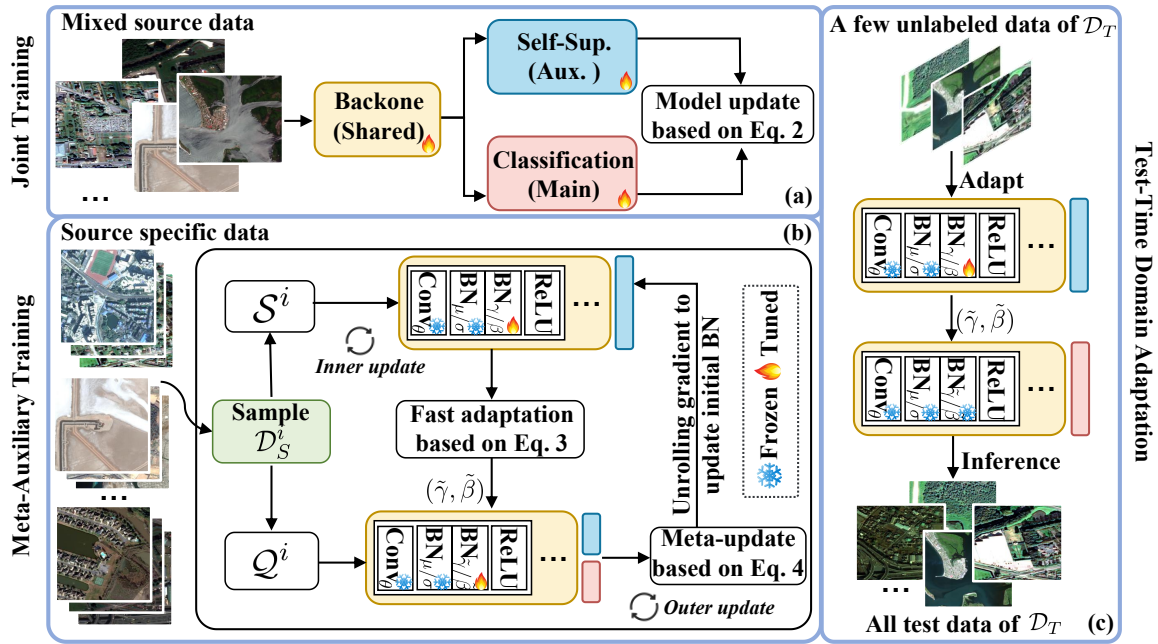
Figure 2: Overview of the proposed MABN. In the joint training stage (a), we train the entire network to learn both label knowledge and normalization statistics by mixing all the source data and performing joint training. During the meta-auxiliary training stage (b), we first obtain the adapted parameters based on the auxiliary loss in the inner loop. Then, the meta-model is updated at the outer loop based on the main task loss computed on adapted parameters. At test-time (c), we simply apply the adaptation step to update the model specifically to an unseen target domain.

et al. 2021; Liu et al. 2022, 2023) reconstruct the input images to achieve internal-learning. ARM (Zhang et al. 2021) incorporates test-time adaptation with DG, which meta-learns a model that is capable of adapting to unseen target domains. Meta-DMoE (Zhong et al. 2022) suggests treating each source domain as an expert, addressing domain shift by querying the relevant knowledge between the target domain data and these experts. However, those methods do not explicitly identify which knowledge and how they should be learned to enlarge the performance gain.

Batch normalization. (Nado et al. 2020) introduce prediction-time BN, utilizing test batch statistics for standardization. Similarly, (Du et al. 2020) and (Hu et al. 2021) update the test statistic using predefined hyper-parameter and moving average, respectively. Conversely, (Schneider et al. 2020) propose to adapt BN statistics by combining source and test batch statistics to mitigate covariate shift. (Lim et al. 2023) interpolate the statistics by adjusting the importance between source and test batch statistics according to the domain-shift sensitivity of each BN layer. The primary difference with existing methods is that instead of learning a perturbable statistic, we focus on investigating the role of affine parameters towards the generalizability of few-shot learners under TT-DA setting.

Meta-learning. Existing meta-learning methods can be categorized into: 1) model-based (Bateni et al. 2020); 2) optimization-based (Ravi and Larochelle 2017); and 3) metric-based (Snell, Swersky, and Zemel 2017). Typi-

cal meta-learning methods utilize bi-level optimization to train a model that is applicable for downstream adaptations. Our work is built upon MAML (Finn, Abbeel, and Levine 2017), which trains a model initialization through episodes of tasks for fast adaptation via gradient updates. Such learning paradigm has been widely applied in different vision tasks, such as zero-shot learning (Wu et al. 2023b) and class incremental learning (Chi et al. 2022; Wu et al. 2023a; Wang et al. 2023). In our case, the adaptation is achieved in an unsupervised manner, and the bi-level optimization is utilized to adapt to unseen domain and generalize well across all data samples in that domain.

The Proposed Method

Problem setting. In this work, we consider the setting in (Zhong et al. 2022) which we refer to as *test-time domain adaptation (TT-DA)*. During the offline learning stage, we have access to M labeled source domains denoted as $\{\mathcal{D}_S^i\}_{i=1}^M$. Each source domain $\mathcal{D}_S^i$ contains a set of label data, i.e. $\mathcal{D}_S^i = (x_S, y_S)^i$ where (x, y) indicate the input image and corresponding label, respectively. After offline training with these source domains, we obtain a trained model and an adaptation mechanism. During testing, we are given a new target domain D_T. Our goal is to adapt the trained model to this target domain using only a small number of unlabeled images from the target domain. Here we assume all the domains share the same label space, but there can be a domain shift between any of these source and target domains $\{D_S^1, D_S^2, ..., D_S^M, D_T\}$.

TT-DA is related to domain adaptation, but there are some key differences. Traditional unsupervised domain adaptation (UDA) assumes access to both source (either single or multi-source) and target domains during training. In contrast, during offline training, TT-DA does not have access to the target domain. During testing, the adaptation of TT-DA cannot access the source domains. Another related setting is source-free domain adaptation (SFDA). The key difference is that the adaptation in SFDA assumes access to a large number of unlabeled data from the target domain. In contrast, TT-DA only requires a small number of target images for the adaptation. TT-DA is a more realistic setting for many real-world scenarios. For example, in wild animal monitoring applications, after we install a surveillance camera in a new location (i.e. domain), we may want to deploy an adapted model after collecting only a few images from the new camera. TT-DA is also related to test-time adaptation (TTA). The key difference is that TTA usually adapts the model for a batch of test examples, then the adapted model is used for prediction on the same batch of examples. In practice, it may not be realistic to adapt the model every time before a prediction is needed. In contrast, TT-DA only adapts the model once using a small batch of images. The adapted model is then used for prediction for all test images of the target domain. In real-world scenarios, this is a more realistic setting.

Motivations

Adapting the model using a few unlabeled data is challenging and the problem is particularly acute when unknown distribution is encountered. Within this complex setting, two fundamental questions require careful consideration: 1) What type of knowledge is most efficacious for adapting to an unseen domain? 2) How can one procure adequate supervision to guide the model's update toward that domain?

Prior work has shown that label and domain knowledge are distinctively encoded in the weight matrix and BN layers, respectively (Li et al. 2016). In the context of TT-DA, where all domains share the same label space, it implicitly indicates that label information can be learned from extensive source data (Li et al. 2016). Consequently, our approach focuses on selectively modulating the BN layers, leaving the well-acquired label knowledge undisturbed. Given a batch of feature map $\mathbf{F}$, BN layer performs normalization and affine transformation separately on every channel as:

$$\mathbf{F}' = \gamma\hat{\mathbf{F}} + \beta, \quad \text{where,} \quad \hat{\mathbf{F}} = \frac{\mathbf{F} - \mu}{\sqrt{\sigma^2 + \epsilon}}. \tag{1}$$

μ and σ are the calculated mean and variance of the batch, and ϵ is a small number to prevent division by 0. γ and β are the two affine parameters. During training, running mean μ' and variance σ' are updated by each batch, and converge to the true statistics of source data. They will be adopted at the inference. However, estimating the statistics of unknown target domains using a few examples is unstable, as the data points are sparsely sampled. To reduce the interference from such instability, we propose to leverage the μ' and σ' computed from source data and only update γ and β. Although the first normalization step will transform the input features towards source distribution, γ and β are optimized to adapt and pull the normalized feature towards target distribution. This approach is straightforward to implement and keeps the minimum number of learnable parameters without introducing any extra inference cost.

To provide supervision for γ and β, and in the meantime, further strengthen the domain knowledge learning, we propose to utilize class-independent self-supervised learning. Specifically, an auxiliary branch is integrated in parallel with the main classification branch as shown in Fig. 2(a). These two branches share the same input feature which is encoded by the backbone. The architecture of the auxiliary branch depends on the SSL algorithm – it can be a single MLP or a more complicated structure. In this work, we do not aim to design novel SSL methods, but rather adopt the existing ones, e.g. BYOL (Grill et al. 2020). After adapting to the target domain, the auxiliary branch can be discarded and only the original network, e.g. ResNet, is retained for inference.

Learning Label-Dependent Representation

Given that the weight matrix encodes rich label information, and all the domains share the same label space, we first perform large-scale training on source data. We mix all the data in $\{\mathcal{D}_S^i\}_{i=1}^{M}$ to uniformly sample the mini-batches. The auxiliary and main branches are updated by optimizing the joint loss:

$$\mathcal{L}_{Joint} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{SSL}. \tag{2}$$

$\mathcal{L}_{SSL}$ and $\mathcal{L}_{CE}$ denote the self-supervised loss and supervised cross-entropy loss, respectively. Note that $\mathcal{L}_{CE}$ can be replaced accordingly (e.g. MSE loss) for regression problems. These two losses are balanced by λ. The running mean μ' and variance σ' statistics collected during the training converge to the accurate statistics of source data.

Learning to Adapt to Unseen Domain Knowledge

The model optimized by $\mathcal{L}_{joint}$ is not necessarily prepared to adapt to unseen domains as the auxiliary loss ($\mathcal{L}_{SSL}$) and the main loss ($\mathcal{L}_{CE}$) are mutually independent. In other words, the parameters updated by the gradient from the auxiliary branch cannot guarantee positive improvement to main task. Moreover, the model lacks awareness of its subsequent learning duty, which involves adapting to unseen domains.

Meta-auxiliary training. To address such an issue, we propose a meta-auxiliary learning scheme to align the gradients between two losses and endow the model to learn to adapt to unseen domains. At the meta-auxiliary training stage, we freeze the weight matrix to preserve the rich label information. We also directly adopt the running mean μ' and variance σ' from the source data to reduce the interference on domain information learning caused by unstable few-shot data. As a result, only the affine parameters $(\gamma, \beta) = \{(\gamma, \beta)^S, (\gamma, \beta)^A\}$ are learnable during this stage. Here the superscripts S and A are used to denote the parameters for the shared backbone and the auxiliary branch, respectively. Note that we adopt a one-layer MLP as the classification head without BN, and $(\gamma, \beta)^A$ are SSL-dependent. If the auxiliary branch has no BN, it can be simply ignored.

To train (γ, β), we employ episodic learning as in meta-learning (Finn, Abbeel, and Levine 2017) to consider every

domain as a "task". We perform nested optimization at the task level instead of instance level, so that (γ, β) are meta-trained to fulfill the task of learning to adapt to unseen domains. For each iteration, we select a source domain (task) $\mathcal{D}_S^i$. We sample an unlabeled support set $\mathcal{S}^i$ and a labeled query set $\mathcal{Q}^i$ from $\mathcal{D}_S^i$. We first adapt (γ, β) to this domain using SSL on the support set $\mathcal{S}^i$ with a learning rate of α as:

$$(\tilde{\gamma}, \tilde{\beta}) = (\gamma, \beta) - \alpha \nabla_{(\gamma, \beta)} \mathcal{L}_{SSL}(\mathcal{S}^i; \theta, (\gamma, \beta)), \quad (3)$$

where θ denotes all the weight matrices. Ideally, $(\tilde{\gamma}, \tilde{\beta})$ should adapt to the domain $\mathcal{D}_S^i$ and improve the main branch. In other words, the adapted model should generalize to all data in that domain in terms of the main task. Therefore, we evaluate $(\tilde{\gamma}, \tilde{\beta})$ on the disjoint labeled query set $\mathcal{Q}^i$ using Eq. 2 as the meta-objective and update as:

$$(\gamma, \beta) \leftarrow (\gamma, \beta) - \delta \nabla_{(\gamma, \beta)} \mathcal{L}_{Joint}(\mathcal{Q}^i; \theta, (\tilde{\gamma}, \tilde{\beta})), \quad (4)$$

where δ is the outer loop learning rate. Updating $(\gamma, \beta)^A$ requires supervision from $\mathcal{L}_{SSL}$, therefore, we use the joint loss. Note, in Eq. 4, the evaluation is conducted on $(\tilde{\gamma}, \tilde{\beta})$ but the gradient update is carried out on original (γ, β) to achieve meta-level updates. The symbol for input and output (x, y) is omitted in Eq. 3 and Eq. 4 for simplicity and the process is repeated until converged. Alg. 1 and Fig. 2(b) elaborate the whole training pipeline.

Meta-auxiliary testing. The meta-parameters (γ, β) have been learned specifically to facilitate the domain knowledge adaptation on unseen target domains. At test-time, given an unseen target domain $\mathcal{D}_T$, the adapted parameters $(\tilde{\gamma}, \tilde{\beta})$ are obtained by simply performing the inner loop as Line 12 in Alg. 1 and Fig. 2(c) using the support set consisting of a few unlabeled images. Then the adapted model can perform inference on all test samples in that domain, and the auxiliary branch is discarded to retain the computational cost.

Experiments

Dataset and evaluation metrics. In this work, follow Meta-DMoE (Zhong et al. 2022) to evaluate our method on five benchmarks from WILDS (Koh et al. 2021): iWild-Cam (Beery et al. 2021), Camelyon17 (Bandi et al. 2018), RxRx1 (Taylor et al. 2019), FMoW (Christie et al. 2018) and PovertyMap (Yeh et al. 2020). Note, that we follow the official training/validation/testing splits, and report the same metrics as in (Koh et al. 2021), including accuracy, Macro F1, worst-case (WC) accuracy, Pearson correlation (r), and its worst-case counterpart. We also evaluate on DomainNet benchmark (Peng et al. 2019). Detailed descriptions of the benchmarks are provided in the supplement.

Model architectures. For faithful comparison, we follow WILDS (Koh et al. 2021) to use ResNet50 (He et al. 2016), DenseNet121(Huang et al. 2017) and ResNet18 (Yeh et al. 2020) as the backbones of our proposed method for iWildCam/RxRx1, Camelyon17/FMoW and PovertyMap datasets. The output of the last average pooling layer of the backbone is served as the input of auxiliary and main branches. As for the auxiliary branch, we select BYOL (Grill

Algorithm 1: Meta-auxiliary training of MABN

Require: α, δ, η: learning rates; B: meta batch size
Require: $\{\mathcal{D}_S^i\}_{i=1}^M$: data of source domains
1: Initialize weight matrix θ and affine params (γ, β)
2: // Learning label representation on mixed source data
3: $(\theta, \gamma, \beta) \leftarrow (\theta, \gamma, \beta) - \eta \nabla_{(\theta, \gamma, \beta)} \mathcal{L}_{Joint}(\mathcal{D}_S; \theta; (\gamma, \beta))$
4: **while** not converged **do**
5:     // Learning to adapt to domain-specific knowledge
7:     Sample a meta batch of B source domains: $\{\mathcal{D}_S^i\}_{i=1}^B$
8:     Reset the loss of current meta batch: $\mathcal{L}_B = 0$
9:     **for** each $\mathcal{D}_S^i$ in $\{\mathcal{D}_S^i\}_{i=1}^B$ **do**
10:         Sample support and query set: $(\mathcal{S}^i, \mathcal{Q}^i) \sim \mathcal{D}_S^i$
11:         // Perform adaptation via self-supervised loss
12:         $(\tilde{\gamma}, \tilde{\beta}) = (\gamma, \beta) - \alpha \nabla_{(\gamma, \beta)} \mathcal{L}_{SSL}(\mathcal{S}; \theta; (\gamma, \beta))$
14:         // Evaluate the adapted $(\tilde{\gamma}, \tilde{\beta})$ using $\mathcal{Q}$ and
15:         // accumulate the loss
16:         $\mathcal{L}_B = \mathcal{L}_B + \mathcal{L}_{Joint}(\mathcal{Q}; \theta; (\tilde{\gamma}, \tilde{\beta})^{S,A})$
17:     **end for**
18:     // Update (γ, β) for current meta batch
19:     $(\gamma, \beta) \leftarrow (\gamma, \beta) - \delta \nabla_{(\gamma, \beta)} \mathcal{L}_B$
21: **end while**

---

et al. 2020) as the self-supervised learning method. Its architecture consists of two MLP layers with BN and ReLU activation, we refer the readers to the original paper for more details. For the main branch, we simply use a linear layer for both classification and regression tasks.

Implementation. We follow (Zhong et al. 2022) to use ImageNet-1K (Deng et al. 2009) pre-trained weights as the initialization to perform joint training. Adam optimizer is used to minimize Eq. 2 with a learning rate (LR) of $1e^{-4}$ for 20 epochs. LR is reduced by a factor of 2 when the loss reaches a plateau. λ in Eq. 2 is set to 0.1. During meta-auxiliary training, we fix the weight matrix of the entire network, and directly use the running statistics μ' and σ' for the BN layers. Only the affine parameters γ and β of the BN layers are further optimized using Alg. 1 for 10 epochs with fixed LR of $3e^{-4}$ for α and $3e^{-5}$ for δ. During testing, for each target domain, we randomly sample 12 images for iWildCam and 32 images for the rest datasets to perform adaptation first (Line 12-13 of Alg. 1). The adapted model is then used to test all the images in that domain. The same process is repeated for all target domains. All the experiments are conducted with 5 random seeds to show the variation.

Experimental Results

Main results. We compare the proposed approach with a variety of methods showing on the WILDS leaderboard, including non-adaptive methods ERM (Vapnik 1999), CORAL (Sun and Saenko 2016), Group DRO (Sagawa et al. 2020), IRM (Arjovsky et al. 2019) and adaptive methods ARM (Zhang et al. 2021), Meta-DMoE (Zhong et al. 2022), PAIR (Chen et al. 2023). The experimental results in Tab. 1 show that the proposed method outperforms all other methods under 8 evaluation metrics across all five datasets. Specifically, our MABN surpasses the most recent method

Methods	iWildCam		Camelyon17	RxRx1	FMoW		PovertyMap	
	Acc	Macro F1	Acc	Acc	WC Acc	Avg Acc	WC Pearson r	Pearson r
ERM	71.6±2.5	31.0±1.3	70.3±6.4	29.9±0.4	32.3±1.25	53.0±0.55	0.45±0.06	0.78±0.04
CORAL	73.3±4.3	32.8±0.1	59.5±7.7	28.4±0.3	31.7±1.24	50.5±0.36	0.44±0.06	0.78±0.05
Group DRO	72.7±2.1	23.9±2.0	68.4±7.3	23.0±0.3	30.8±0.81	52.1±0.5	0.39±0.06	0.75±0.07
IRM	59.8±3.7	15.1±4.9	64.2±8.1	8.2±1.1	30.0±1.37	50.8±0.13	0.43±0.07	0.77±0.05
ARM-CML	70.5±0.6	28.6±0.1	84.2±1.4	17.3±1.8	27.2±0.38	45.7±0.28	0.37±0.08	0.75±0.04
ARM-BN	70.3±2.4	23.7±2.7	87.2±0.9	31.2±0.1	24.6±0.04	42.0±0.21	0.49±0.21	**0.84±0.05**
ARM-LL	71.4±0.6	27.4±0.8	84.2±2.6	24.3±0.3	22.1±0.46	42.7±0.71	0.41±0.04	0.76±0.04
Meta-DMoE	77.2±0.3	34.0±0.6	91.4±1.5	29.8±0.4	35.4±0.58	52.5±0.18	0.51±0.04	0.80±0.03
PAIR	74.9±1.1	27.9±0.9	74.0±7.2	28.8±0.0	35.4±1.30	-	0.47±0.09	-
MABN (ours)	**78.4±0.6**	**38.3±1.2**	**92.4±1.9**	**32.7±0.2**	**36.6±0.41**	**53.2±0.52**	**0.56±0.05**	**0.84±0.04**

Table 1: Comparison with the state-of-the-arts on the WILDS benchmark under the out-of-distribution setting. Metric means and standard deviations are reported across replicates. For each comparison method, we either obtain its result directly reported in the leaderboard or use the released official model with default parameters.

Method	clip	info	paint	quick	real	sketch	avg
ARM	49.7	16.3	40.9	9.4	53.4	43.5	35.5
Meta-DMoE	63.5	21.4	51.3	14.3	62.3	52.4	44.2
Ours	**64.2**	**23.6**	**51.5**	**15.2**	**64.6**	**54.1**	**45.5**

Table 2: Comparison on the DomainNet (Accuracy).

Adapted $(\tilde{\gamma}, \tilde{\beta})$	No adapt	Not-matched	Matched
Accuracy	74.69	72.39	**78.40**
Macro-F1	36.77	33.32	**38.27**

Table 3: Verification of domain knowledge learning. "No adapt" means the meta-learned (γ, β) are used for all domains without adaptation. "Not matched" means each target domain randomly uses the adapted $(\tilde{\gamma}, \tilde{\beta})$ from other domains instead of its own. "Matched" means each target domain uses its own adapted $(\tilde{\gamma}, \tilde{\beta})$.

Method	Update BN		Update Affine	
	Acc	Macro F1	Acc	Macro F1
TENT (min. entropy)	33.27	0.77	75.92	36.40
Our (min. auxiliary)	75.86	36.76	78.40	38.27
Our+TENT	75.84	31.93	79.68	38.85

Table 4: Comparison and integration of our MABN with other TTA method. TENT learns more label-related knowledge by minimizing entropy, which can be seamlessly combined with our method to achieve superior performance.

PAIR in classification accuracy by 3.5%, 18.4% and 3.9% on iWildCam, Camelyon17 and RxRx1 datasets. We also outperform Meta-DMoE by a (%) margin of 1.2/4.6 on iWildCam, 1.0 on Camelyon17, 2.9 on RxRx1, 1.2/0.7 on FMoW and 0.05/0.04 on PovertyMap. Note, our method only uses a single model which is more lightweight while Meta-DMoE has a group of teacher models. These results show the effectiveness of identifying the crucial parameters for adapting to domain-specific knowledge. Such adaptability is important for enhancing the generalization of the target domain. Tab. 2 reports the superiority of our method on DomainNet.

Is MABN really learning the domain knowledge? To show that our adapted models have learned domain-specific information and adapt to each of the target domains, we conduct an experiment to shuffle the adapted $(\tilde{\gamma}, \tilde{\beta})$ across target domains. Specifically, assume there are N target domains, we save all the adapted affine parameters as $\{(\tilde{\gamma}, \tilde{\beta})_j\}_{j=1}^N$. For j^{th} target domain, we randomly use $(\tilde{\gamma}, \tilde{\beta})_u$ which be-

longs to other domains (i.e., $i \neq u$), we name it "Not matched". We then compare it with when the jth target domain utilizes its own $(\tilde{\gamma}, \tilde{\beta})_j$, which we refer to as "Matched". As reported in Tab. 3, several observations can be made: 1) when the target domains utilize non-matched adapted domain knowledge, the performance dropped significantly. However, the performance boosts when the target domain uses its own knowledge; 2) Using the correct adapted $(\tilde{\gamma}, \tilde{\beta})$, immense improvement is made compared to the non-adaptive model. Thus, we can conclude that our method learns unique domain knowledge for every target domain and improves the overall performance. We also visualize the features before and after adaptation using t-SNE (Van der Maaten and Hinton 2008). As shown in Fig. 3, each class cluster is more discriminative after adaptation.

To further show that our method is learning domain knowledge, we seamlessly integrate MABN with an entropy-based TTA method TENT (Wang et al. 2021), which is more label-related. To do so, for each target domain, we first use MABN to adapt the domain and then apply TENT on each batch. We also compare the case of taking the whole BN layer and only the affine parameters. Several conclusions can be drawn from Tab. 4: 1) Adapting the entire BN layer proves to be problematic, as the normalization is unstable and interferes with the affine parameters; 2) Our method outperforms TENT due to the fact that label-related information is already contained within the weight matrix,
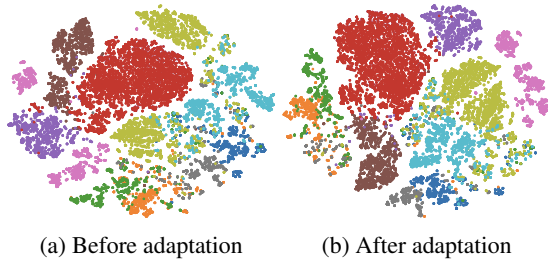
(a) Before adaptation   (b) After adaptation

Figure 3: t-SNE visualization of features before and after adaptation. Each data sample is represented as a point, and each color corresponds to a class randomly selected from the target domain of the iWildCam dataset.

Index	SSL	$Param.$	TS	$Adapt$	iWildCam Acc	F1
1	✗	All	CE	✗	68.7	31.3
2	✓	All	Joint	✗	70.5	33.2
3	✓	BN	Joint	✓	68.2	30.5
4	✓	Aff	Joint	✓	71.1	33.9
5	✓	All	Meta	✓	72.0	29.4
6	✓	Aff	Meta	✗	74.7	36.8
7	✓	Aff	Meta	✓	78.4	38.3

Table 5: Ablation studies for different components of our framework. SSL denotes SSL branch. $Param.$ denotes which parameters are updating, including the whole network ("All"), BN layer ("BN") or only the affine parameters ("Aff"). TS denotes the training scheme. $Adapt$ denotes whether adapting to each target domain.

thereby necessitating the learning of domain-specific information; 3) Since we aim to learn the domain information, integrating with TENT further pushes the boundary by a large margin (1.28% and 0.58% on accuracy and F1 score).

Ablation studies. To evaluate the contribution from each component in MABN on the iWildCam dataset. As reported in Tab. 5, several conclusions can be drawn: 1) (Index 1 vs. 2) adding the SSL branch and jointly training is beneficial to handle distribution shift. The hypothesis is that SSL improves the discrimination of the learned feature; 2) (Index 2 vs. 3) adapting the model by updating the whole BN even degrades the performance as the normalization statistic learned from a few images is unstable. However, freezing the normalization and only updating the affine parameters overturns the results (Index 4) and positive improvement is observed. The results confirm our hypothesis: the unstable normalization statistics can interfere the adaptation process and the affine parameters alone are more effective to adapt to unseen domains; 3) Our meta-auxiliary training aims to meta-learn the affine parameters that can adapt to any domains effectively. It implicitly indicates that the meta-learned affine parameters are more general to all distributions instead of the one only fit to source domains. Therefore, even without adaptation, the meta-learned model achieves better generalization (Index 4 vs. 6). Besides, our meta-objective aligns

Self-supervised	Backbone	Training	iWildCam Acc	F1
None (baseline)	ResNet50	CE	68.7	31.3
Rotation (Sun et al. 2020)	ResNet50	Joint	69.2	31.5
Rotation (Sun et al. 2020)	ResNet50	Meta	72.8	33.0
MAE (He et al. 2022)	ViT-Base	Joint	71.7	33.8
MAE (He et al. 2022)	ViT-Base	Meta	74.9	35.1
Ours (BYOL)	ResNet50	Joint	70.5	33.2
Ours (BYOL)	ResNet50	Meta	78.4	38.3

Table 6: Evaluation on different SSL tasks. Note, we adapt ViT-Base by updating the last block as it does not have BN layer, verifying the effectiveness of meta-auxiliary training.

two learning objectives, so that the adapted affine parameters by SSL to target domains further push the performance boundary (Index 7); 4) But without properly disentangling the learning knowledge and identifying the adaptive parameters, it is ineffective to adapt to unseen domains (Index 5).

Hyper-parameter analysis. We study the performance of our MABN under various parameter settings. We first investigate the sensitivity on the number of unlabeled data used for adaptation at test-time. Intuitively, more unlabeled data from a target domain reveals a more precise representation of its underlying distribution. Consequently, as reported in Fig. 6 in the supplementary, increasing the number of unlabeled data empowers MABN to more effectively exploit the domain-specific knowledge. Moreover, in the extreme case with only one unlabeled data, MABN still performs better than without adaptation. These results again indicate that adapting with only affine parameters is more effective by creating less ambiguity from unstable statistics. We also evaluate MABN with different SSL tasks as in Tab. 6. It is observed that different SSL tasks affect MABN. However, they are all superior over the baseline. On the other hand, our meta-auxiliary learning scheme is able to improve over joint training for all SSL methods even with ViT (Dosovitskiy et al. 2020), in which the last layer is updated as it does not have BN layers. It is highlighted that the proposed MABN is general and serves as a plug-and-play module for existing methods to improve generalization.

Conclusion

In this paper, we have presented a simple yet effective method MABN for test-time domain adaptation. Specifically, our proposed method achieves domain knowledge learning by only adapting the affine parameters in BN. To further extract domain knowledge from a limited amount of unlabeled data, we establish an auxiliary branch with label-independent self-supervised learning. In addition, we introduce a meta-learning-based optimization strategy so that the affine parameters are learned in a way that facilitates domain knowledge adaptation. Extensive experiments on various real-world domain shift benchmarks demonstrate the superiority of MABN over the state-of-the-art.

Acknowledgments

References

Arjovsky, M.; Bottou, L.; Gulrajani, I.; and Lopez-Paz, D. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Bandi, P.; Geessink, O.; Manson, Q.; Van Dijk, M.; Balkenhol, M.; Hermsen, M.; Bejnordi, B. E.; Lee, B.; Paeng, K.; Zhong, A.; et al. 2018. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 38(2): 550–560.

Bateni, P.; Goyal, R.; Masrani, V.; Wood, F.; and Sigal, L. 2020. Improved few-shot visual classification. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Beery, S.; Agarwal, A.; Cole, E.; and Birodkar, V. 2021. The iwildcam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*.

Bronskill, J.; Gordon, J.; Requeima, J.; Nowozin, S.; and Turner, R. 2020. Tasknorm: Rethinking batch normalization for meta-learning. In *International Conference on Machine Learning*. PMLR.

Chen, Y.; Zhou, K.; Bian, Y.; Xie, B.; Wu, B.; Zhang, Y.; Ma, K.; Yang, H.; Zhao, P.; Han, B.; and Cheng, J. 2023. Pareto Invariant Risk Minimization: Towards Mitigating the Optimization Dilemma in Out-of-Distribution Generalization. In *International Conference on Learning Representations*.

Chi, Z.; Gu, L.; Liu, H.; Wang, Y.; Yu, Y.; and Tang, J. 2022. Metafscil: A meta-learning approach for few-shot class incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Chi, Z.; Wang, Y.; Yu, Y.; and Tang, J. 2021. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Choi, Y.; Choi, M.; Kim, M.; Ha, J.-W.; Kim, S.; and Choo, J. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Christie, G.; Fendley, N.; Wilson, J.; and Mukherjee, R. 2018. Functional map of the world. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Du, Y.; Zhen, X.; Shao, L.; and Snoek, C. G. 2020. Metanorm: Learning to normalize few-shot batches across domains. In *International Conference on Learning Representations*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1): 2096–2030.

Geirhos, R.; Temme, C. R.; Rauber, J.; Schütt, H. H.; Bethge, M.; and Wichmann, F. A. 2018. Generalisation in humans and deep neural networks. *Advances in Neural Information Processing Systems*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *IEEE Conference on Computer Vision and Pattern Recognition*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Hu, X.; Uzunbas, G.; Chen, S.; Wang, R.; Shah, A.; Nevatia, R.; and Lim, S.-N. 2021. Mixnorm: Test-time adaptation through online normalization estimation. *arXiv preprint arXiv:2110.11478*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Gao, I.; et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*.

Li, R.; Jiao, Q.; Cao, W.; Wong, H.-S.; and Wu, S. 2020. Model adaptation: Unsupervised domain adaptation without source data. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Li, Y.; Wang, N.; Shi, J.; Liu, J.; and Hou, X. 2016. Revisiting batch normalization for practical domain adaptation. In *International Conference on Learning Representations*.

Liang, H.; Quader, N.; Chi, Z.; Chen, L.; Dai, P.; Lu, J.; and Wang, Y. 2022. Self-supervised spatiotemporal representation learning by exploiting video continuity. In *AAAI Conference on Artificial Intelligence*.

Lim, H.; Kim, B.; Choo, J.; and Choi, S. 2023. TTN: A domain-shift aware batch normalization in test-time adaptation. In *International Conference on Learning Representations*.

Liu, H.; Chi, Z.; Yu, Y.; Wang, Y.; Chen, J.; and Tang, J. 2023. Meta-Auxiliary Learning for Future Depth Prediction in Videos. In *IEEE Winter Conference on Applications of Computer Vision*.

Liu, H.; Wu, Z.; Li, L.; Salehkalaibar, S.; Chen, J.; and Wang, K. 2022. Towards multi-domain single image dehazing via test-time training. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2018. Conditional adversarial domain adaptation. *Advances in Neural Information Processing Systems*.

Lv, F.; Liang, J.; Li, S.; Zang, B.; Liu, C. H.; Wang, Z.; and Liu, D. 2022. Causality inspired representation learning for domain generalization. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Muandet, K.; Balduzzi, D.; and Schölkopf, B. 2013. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*.

Nado, Z.; Padhy, S.; Sculley, D.; D'Amour, A.; Lakshminarayanan, B.; and Snoek, J. 2020. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*.

Pei, Z.; Cao, Z.; Long, M.; and Wang, J. 2018. Multi-adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence*.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision*.

Qiu, Z.; Zhang, Y.; Lin, H.; Niu, S.; Liu, Y.; Du, Q.; and Tan, M. 2021. Source-free domain adaptation via avatar prototype generation and adaptation. In *International Joint Conference on Artificial Intelligence*.

Ravi, S.; and Larochelle, H. 2017. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*.

Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2020. Distributionally Robust Neural Networks. In *International Conference on Learning Representations*.

Schneider, S.; Rusak, E.; Eck, L.; Bringmann, O.; Brendel, W.; and Bethge, M. 2020. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*.

Seo, S.; Suh, Y.; Kim, D.; Kim, G.; Han, J.; and Han, B. 2020. Learning to optimize domain specific normalization for domain generalization. In *European Conference on Computer Vision*.

Shu, R.; Bui, H. H.; Narui, H.; and Ermon, S. 2018. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*.

Standley, T.; Zamir, A.; Chen, D.; Guibas, L.; Malik, J.; and Savarese, S. 2020. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*.

Sun, B.; and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision Workshops*.

Sun, Y.; Wang, X.; Liu, Z.; Miller, J.; Efros, A.; and Hardt, M. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*.

Taylor, J.; Earnshaw, B.; Mabey, B.; Victors, M.; and Yosinski, J. 2019. Rxrx1: An image set for cellular morphological variation across many experimental batches. In *International Conference on Learning Representations*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vapnik, V. N. 1999. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2021. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*.

Wang, J.; Feng, S.; Lyu, G.; and Gu, Z. 2023. Triple-Granularity Contrastive Learning for Deep Multi-View Subspace Clustering. In *Proceedings of the ACM International Conference on Multimedia*, 2994–3002.

Wang, M.; and Deng, W. 2018. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153.

Wu, Y.; Chi, Z.; Wang, Y.; and Feng, S. 2023a. Metagcd: Learning to continually learn in generalized category discovery. In *IEEE International Conference on Computer Vision*.

Wu, Y.; Liang, T.; Feng, S.; Jin, Y.; Lyu, G.; Fei, H.; and Wang, Y. 2023b. MetaZSCIL: A Meta-Learning Approach for Generalized Zero-Shot Class Incremental Learning. In *AAAI Conference on Artificial Intelligence*.

Wu, Y.; Liu, H.; Feng, S.; Jin, Y.; Lyu, G.; and Wu, Z. 2021. GM-MLIC: graph matching based multi-label image classification. In *International Joint Conference on Artificial Intelligence*.

Yeh, C.; Perez, A.; Driscoll, A.; Azzari, G.; Tang, Z.; Lobell, D.; Ermon, S.; and Burke, M. 2020. Using publicly available satellite imagery and deep learning to understand economic well-being in Africa. *Nature Communications*, 11(1): 2583.

Zhang, M.; Marklund, H.; Dhawan, N.; Gupta, A.; Levine, S.; and Finn, C. 2021. Adaptive risk minimization: Learning to adapt to domain shift. *Advances in Neural Information Processing Systems*.

Zhong, T.; Chi, Z.; Gu, L.; Wang, Y.; Yu, Y.; and Tang, J. 2022. Meta-dmoe: Adapting to domain shift by meta-distillation from mixture-of-experts. *Advances in Neural Information Processing Systems*.

Zhou, K.; Yang, Y.; Hospedales, T.; and Xiang, T. 2020. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*.