Overcoming Domain Knowledge Forgetting in Continual Test-Time Adaptation via Siamese Networks

Zhihong Xu XUZHH2022@MAIL.SIM.AC.CN
School of Information Science and Technology, ShanghaiTech University, Shanghai, China; Bio-vision System Laboratory, Science and Technology on Micro-system Laboratory, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China

Wenjun Shi WJS@MAIL.SIM.AC.CN
Donchen Zhu* DCHZHU@MAIL.SIM.AC.CN
Xiaolin Zhang XLZHANG@MAIL.SIM.AC.CN
Lei Wang WANGL@MAIL.SIM.AC.CN
Jiamao Li JMLI@MAIL.SIM.AC.CN
Bio-vision System Laboratory, Science and Technology on Micro-system Laboratory, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China; University of Chinese Academy of Sciences

Editors: Hung-yi Lee and Tongliang Liu

Abstract

Test-Time Adaptation (TTA) requires adapting a source-domain model to the target domain using online test data inputs. Existing methods that focus on adjusting normalization layers to swiftly adapt to a new domain often neglect the problem of domain knowledge forgetting, which hinders the model's generalization capability. To address this, we propose a novel Anti-forgetting Test-time Adaptation Network (ATAN) which consists of three Siamese networks—Forerunner, Bridge and Momentum. The bridge network transfers domain-specific knowledge from the forerunner network to the momentum network which effectively overcomes forgetting by integrating cross-domain knowledge. To further enhance the adaptability of the forerunner network, we propose reconstructing its loss function based on the voting information from the Siamese networks. To strengthen the learning of domain-invariant features, we introduce a weak augmentation consistency loss for the bridge network. Extensive experiments on corruption and natural shift datasets demonstrate the effectiveness and generalization of ATAN in long-term test-time domain adaptation scenarios.

Keywords: Domain Adaptation; Test-time adaptation; Anti-forgetting

1. Introduction

Deep neural networks perform exceptionally well in computer vision tasks when training and test data share the same distribution (He et al., 2016; Dosovitskiy et al., 2020; Liu et al., 2024b). However, post-deployment factors such as weather, lighting, and sensor conditions can induce dynamic changes in the target distribution. Under such circumstances, a model trained on the source domain may suffer significant performance drops in the target domain (Taori et al., 2020). Domain adaptation (Tzeng et al., 2014) aims to reduce the distribution gap between the source and target domains in the feature space during the training phase, thereby obtaining a highly generalized static model to address this issue. It typically

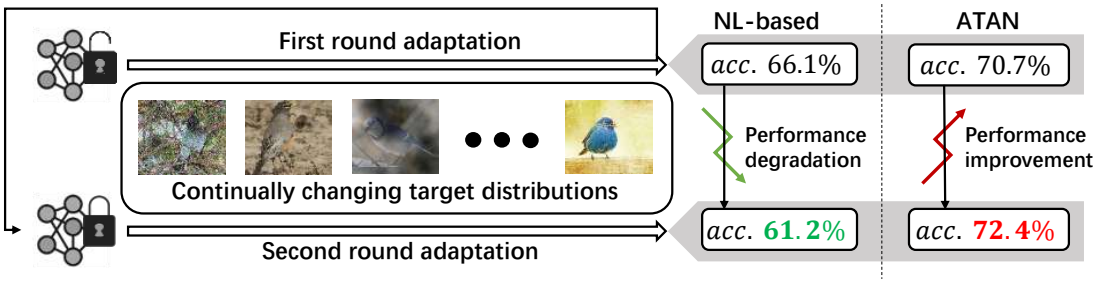

Figure 1: Once the adapted model is frozen, normalization layer-based (NL-based) methods often exhibit noticeable performance degradation in previously encountered domains, indicating the occurrence of domain knowledge forgetting. The proposed method, ATAN, leveraging the momentum network that integrates multi-domain knowledge, significantly overcomes forgetting.

requires a large amount of source domain data during training. However, source domain data is sometimes inaccessible due to commercial or privacy concerns (Wang et al., 2020). Furthermore, static models inherently face bottlenecks in generalization ability (Hendrycks and Dietterich, 2019a). In scenarios requiring real-time predictions for dynamically changing target domain data, traditional domain adaptation approaches are often inadequate. To address this, test-time adaptation has been proposed, which adjusts the parameters of a pretrained model or corrects its outputs (Marsden et al., 2024) using target domain data received during the testing phase to adapt to the new domain.

Existing mainstream methods for test-time adaptation (TTA) can be divided into two categories. **a)** Normalization layer-based (NL-based) methods (Wang et al., 2020; Niu et al., 2022, 2023) adjust the parameters of the normalization layers through entropy reduction to achieve domain-specific adaptation. However, in the process of adapting to a new domain, these methods forget knowledge from the previous domains. Specifically, they can perform well in the current domain, but once the model is frozen, their performance on previously seen domains may significantly decline (Fig. 1). **b)** The other category is self-distillation-based methods (Wang et al., 2022; Yuan et al., 2023; Döbler et al., 2023) which leverage a mean teacher to provide robust pseudo-labels for the student model. After prolonged adaptation, these approaches can achieve good generalization performance. However, due to the large momentum parameter in the exponential moving average process, these methods tend to perform poorly in the early stages of adaptation. In other words, such methods exhibit a certain degree of performance lag during the early stages of domain adaptation, which diminishes their overall usability.

To enable dexterous adaptation while overcoming domain knowledge forgetting, we propose a method called ATAN (Anti-forgetting Test-time Adaptation Network). ATAN consists of three Siamese networks. **Forerunner Network**: Deployed with NL-based (Normalization-layer) methods, it updates the normalization layer parameters through an entropy reduction objective, thereby deftly learning domain-specific knowledge of the new target domain. **Bridge Network**: Using the outputs of the forerunner network as super-

vision to update all parameters, it transfers the domain knowledge from the normalization layers of the forerunner network to all the trainable parameters. **Momentum Network**: Constructed as the exponential moving average of the bridge network, it effectively integrates multi-domain knowledge during the adaptation process. Within this framework, ATAN leverages the forerunner network for domain-specific adaptation and utilizes the momentum network to mitigate forgetting.

Furthermore, we propose a vote-based loss adjustment mechanism to improve the adaptability of the forerunner network. Specifically, we assess the reliability of samples based on the voting consistency of the Siamese networks and reconstruct the loss function of the forerunner network, thereby reducing the error accumulation. Finally, to enhance the learning of domain-invariant features, we construct a weak augmentation consistency loss for the bridge network, ensuring that its predictions on weakly augmented samples align with the predictions of the forerunner network and the momentum network on the original samples.

Our primary contributions can be summarized as follows:

- A novel Anti-forgetting Test-time Adaptation Network (ATAN) is proposed to overcome domain knowledge forgetting by transferring the domain-specific knowledge from a forerunner network to a bridge network and ultimately integrating it into a momentum network. Extensive experiments on corruption and natural shift datasets validate the effectiveness of our ATAN.
- A vote-based loss adjustment mechanism is proposed, which ingeniously utilizes the consistency of Siamese networks to filter out reliable samples and noise samples, thereby reducing the error accumulation and enabling more effective model updates.
- A weak augmentation consistency loss is constructed between the bridge network, forerunner network, and momentum network, which enhances the model's ability to learn domain-invariant features.

2. Related Work

Unsupervised Domain Adaptation (UDA) utilizes labeled source domain data and unlabeled target domain data to achieve better generalization performance of the model over the target domain. A common category of UDA methods focuses on aligning the source and target domain distributions in the feature space. This includes introducing domain discriminators for adversarial learning (Long et al., 2018), constructing loss functions based on domain discrepancies (Long et al., 2015). In addition, some methods assist training at the input level by generating labeled target domain samples (Zhu et al., 2017), while others improve predictions through regularization at the output level (Chen et al., 2019). Although these methods have significantly advanced domain adaptation research, they all require simultaneous access to both source and target domain data during the adaptation process. Moreover, good generalization performance can only be achieved after prolonged, multi-epoch training.

Test-Time Adaptation (TTA) typically operates under the premise of inaccessible source data. The primary research motivation of TTA is to adjust the model or its outputs during test time to adapt to the target domain distribution. Numerous studies (Wang et al.,

2020; Niu et al., 2022) focus on Fully Test-Time Adaptation (FTTA), which assumes that adaptation is required for a single, static target domain distribution at a time. Tent (Wang et al., 2020) first proposed updating the Batch Normalization (BN) layers through entropy minimization to achieve test-time adaptation. EATA (Niu et al., 2022) reduces error accumulation and catastrophic forgetting in Tent (Wang et al., 2020) through sample filtering and Fisher regularization. T3A achieves gradient-free TTA through a dynamically updated prototype classifier. Some studies (Zhang et al., 2022; Niu et al., 2023; Gong et al., 2022) consider more complex test data stream scenarios. MEMO (Zhang et al., 2022) focuses on single-sample inputs, applying multiple weak augmentations to each sample. The model is optimized by minimizing the average prediction entropy across these augmented samples. SAR (Niu et al., 2023) addresses inputs that are domain-mixed, mini-batched, and label-imbalanced, optimizes the model using more stable Group Normalization and Layer Normalization, along with a sharpness-aware entropy minimization loss. Methods designed for FTTA sometimes fail to adapt to dynamically changing target domain distributions (Wang et al., 2022). Consequently, recent research has increasingly focused on Continual Test-Time Adaptation (CTTA), or considers it as one of the essential scenarios to be addressed (Wang et al., 2022; Döbler et al., 2023; Gan et al., 2023).

Continual Test-Time Adaptation (CTTA) focuses on adapting the source model to continuously shifting target domain distribution. CoTTA (Wang et al., 2022) first defined the CTTA setting and introduced mean teacher self-distillation (Tarvainen and Valpola, 2017) to address this task; additionally, it mitigates catastrophic forgetting through parameter probabilistic reset and enhances robust pseudo-label generation through averaging. Some methods also adopt self-distillation as the framework but consider more flexible and complex scenarios (Döbler et al., 2023; Yuan et al., 2023). RMT (Döbler et al., 2023) flexibly considers scenarios where the source domain is available in reality, utilizing source prototypes to compute losses and addressing catastrophic forgetting through source replay. To address potential imbalanced data stream issues in the CTTA task, RoTTA (Yuan et al., 2023) employs a dynamic repository to compute global robust statistics. In dynamic scenarios, the mean teacher in self-distillation may generate noisy pseudo-labels, resulting in error accumulation in the model. To address this issue, ADMA (Liu et al., 2024a) uses a masked autoencoder to mask and reconstruct regions with high distribution shift in images, enabling better learning of domain-invariant features; VDP (Gan et al., 2023) fundamentally avoids error accumulation problem by reformulating the input data instead of adjusting the parameters of source model. Due to the lack of research on Universal TTA, ROID (Marsden et al., 2024) achieves a superior method across various datasets and base models through the design of a detailed loss weighting strategy, weight ensembling with source parameters, and prior correction. These studies have significantly advanced the progress of CTTA. However, there is still a lack of explicit discussion and solutions regarding the problem of domain knowledge forgetting.

3. Method

3.1. Domain Knowledge Forgetting in CTTA

CTTA definition. At the outset, there is only one pre-trained model f_{θ_0} from the source domain $D^S = (X^{\mathrm{S}}, Y^{\mathrm{S}})$, where θ_0 represents the pre-trained parameters. Our objective is

to adapt f_{θ_0} to continuously changing target domains $D^T = \{D_1^T, D_2^T, \cdots, D_n^T\}$. At time step t, a batch of test data $\mathbf{x}^t$ is input. We need to utilize $\mathbf{x}^t$ to adjust the model parameters from θ_t to θ_{t+1} with the goal of improving the model's predictive performance on $\mathbf{x}^{t+1}$. It should be noted that the source data is inaccessible, and the test data is accessed online.

In this work, we also aim for the adapted model f_{θ_n} to effectively retain knowledge from previously encountered domains. Specifically, when the model is frozen, ensure that no performance degradation or knowledge forgetting occurs on $D_1^T \rightarrow D_{n-1}^T$.

Domain knowledge forgetting. NL-based methods have made significant progress in recent years (Wang et al., 2020; Niu et al., 2022; Marsden et al., 2024) but face a critical limitation: while they can deftly adapt to new domain distributions, they may fail to retain knowledge from previously encountered domains, a phenomenon we define as domain knowledge forgetting. As a result, even after extended adaptation, the model's true generalization capability cannot be fundamentally improved. To quantify the degree of domain knowledge forgetting, we propose a metric called the **mean forgetting ratio(MFR)**. Specially, we conduct two rounds of testing on CTTA datasets. During the first round, the model performed normal test-time domain adaptation. Afterward, the adapted model is frozen and subjected to a second round of testing. MFR measures the discrepancy in error rates of a model across two rounds of testing (before and after freezing).

$$\mathrm{MFR} = \frac{\frac{1}{U}\sum_{i=1}^{U} err_i^{\mathrm{1st}} - \frac{1}{U}\sum_{i=1}^{U} err_i^{\mathrm{2nd}}}{\frac{1}{U}\sum_{j=1}^{U} err_j^{\mathrm{1st}}} = \frac{\sum_{i=1}^{U}\left(err_i^{\mathrm{1st}} - err_i^{\mathrm{2nd}}\right)}{\sum_{j=1}^{U} err_j^{\mathrm{1st}}} \tag{1}$$

where U denotes the number of domains, err_i^{1st} and err_i^{2nd} represent the error rates in the first and second round of testing for the i-th domain, respectively. If the MFR is negative, it indicates that the error rate in the second test is higher than that in the first, signifying the occurrence of domain knowledge forgetting. Conversely, a positive MFR suggests that the model not only avoids forgetting but also leverages knowledge from subsequent domains to improve predictions in previous domains. MFR is similar to the backward transfer metric Lopez-Paz and Ranzato (2017) in continual learning, but the former focuses on the impact of adapting to new domains on performance in old domains (where the task remains the same), while the latter emphasizes the impact of learning new tasks on performance in old tasks.

The occurrence of domain knowledge forgetting in NL-based methods may be attributed to the following reasons: a) The parameters of the normalization layers are too limited to accommodate knowledge from multiple domains simultaneously. For example, in ResNet50, the parameters of all normalization layers account for only 0.21% of the total. b) When directly updating the model through backpropagation, sharp gradients can lead to excessively large adjustments in parameters (Niu et al., 2023), making it challenging to maintain temporal consistency. The following sections will detail how our proposed method, ATAN, builds upon NL-based approaches to overcome the issue of domain knowledge forgetting.

3.2. The Structure of the Proposed ATAN

ATAN consists of three Siamese networks, the forerunner network, the bridge network, and the momentum network, all initialized from the same pretrained model.

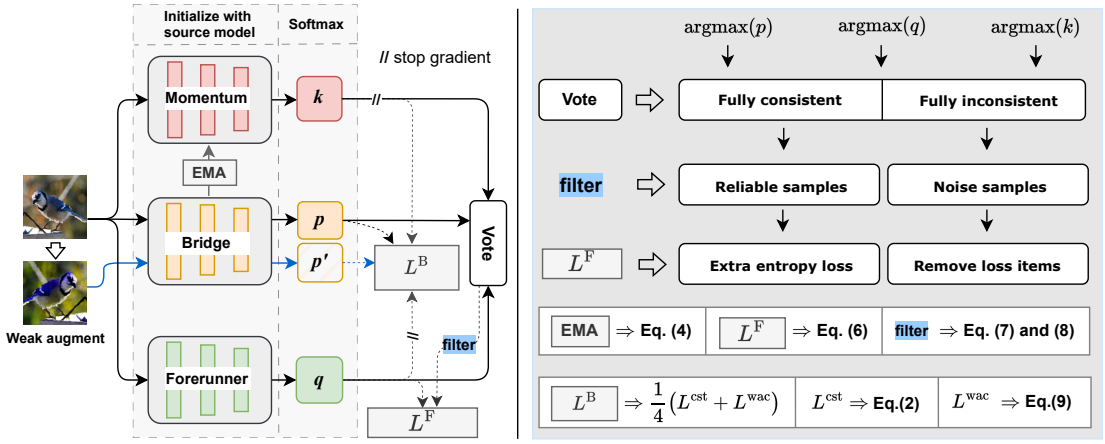

Figure 2: An overview of the proposed ATAN framework. The bridge network aligns with both the forerunner and the momentum network on regular samples and weakly augmented samples. The voting results of the three networks are utilized to calculate the final prediction and guide the loss calculation for the forerunner network.

The forerunner network is deployed with a NL-based method, which serves to deftly adapt to the new distribution by updating the parameters of the normalization layers. The role of normalization layers in domain adaptation and the domain-specific adaptation capability of NL-based methods have been validated in many previous works (Li et al., 2018; Wang et al., 2020; Marsden et al., 2024).

The bridge network updates all its parameters under the supervision of the outputs from the forerunner and the momentum networks. Its functions include: a) Radiating the new domain knowledge learned by the forerunner network from the normalization layers to all trainable layers through consistency loss, thereby increasing the capacity for knowledge storage. b) Acting as a bridge between the forerunner network and the momentum network, transferring new domain knowledge from the former to the latter. The bridge network computes the loss using the outputs of both the forerunner network and the momentum network as supervision.

$$L^{\text{cst}} = L^{\text{sce}}(p, q) + L^{\text{sce}}(p, k) \tag{2}$$

where p, q, k represent the Softmax outputs of the bridge network, forerunner network, and momentum network, respectively.

$$L^{\text{sce}}(p, q) = -\sum_{c=1}^{C}(p_c \log(q_c) + q_c \log(p_c)) \tag{3}$$

where C represents the number of classes. $L^{\text{sce}}(p, k)$ is defined similarly. The reason for introducing the momentum network as supervision is to maintain the stability of the bridge network's features, as the momentum network approximatively represents a temporal average of the bridge network.

The momentum network is constructed as an exponential moving average (EMA) of the bridge network.

$$\theta'_{t+1} = \alpha\theta'_t + (1-\alpha)\theta_{t+1} \tag{4}$$

where θ', θ represent the parameters of the momentum network and the bridge network, respectively. α is the momentum coefficient, and t indicates the time step. The primary function of the momentum network is to continuously integrate domain knowledge originating from the forerunner network during adaptation, preventing knowledge waste and overcoming domain knowledge forgetting.

In ATAN, the three networks are updated in different ways, potentially learning distinct feature views (Allen-Zhu and Li, 2020). By averagining the outputs of them, we can leverage multi-view information to enhance the robustness of the prediction.

$$\hat{y} = \arg\max_c (p + q + k) \tag{5}$$

An overview of the ATAN framework is shown in Fig. 2.

3.3. Model Updates Guided by Voting Consistency

Inspired by the idea of collaborative training of multiple classifiers in Tri-Training (Zhou and Li, 2005), we propose to assess the reliability of sample prediction based on the voting consistency of Siamese networks in ATAN. Samples with fully consistent or fully inconsistent voting results from the three networks can be regarded as high-confidence samples and noise samples, respectively. When calculating the loss function for the forerunner network, the loss terms corresponding to noise samples are first removed to reduce error accumulation caused by incorrect predictions (for NL-based methods with a built-in sample selection step, we skip this procedure to avoid disrupting their original selection logic). Next, an additional entropy loss is computed for high-confidence samples and incorporated into the final loss function, guiding the forerunner network to update in a more reliable direction.

$$L^{\mathrm{F}} = \frac{1}{N - |S|} \sum_{i=1, i\notin S}^{N} l_i^{\mathrm{F}} + \frac{1}{N} \sum_{j \in R} h_j^{\mathrm{F}} \tag{6}$$

$$S = \left\{ i | 1 \le i \le N, \begin{array}{r} \arg\max_c(p^{(i)}) \neq \arg\max_c(q^{(i)}), \\ \arg\max_c(p^{(i)}) \neq \arg\max_c(k^{(i)}), \\ \arg\max_c(q^{(i)}) \neq \arg\max_c(k^{(i)}) \end{array} \right\} \tag{7}$$

$$R = \left\{ j | 1 \le j \le N, \begin{array}{r} arg\max_c(p^{(j)}) = \arg\max_c(q^{(j)}) \\ = \arg\max_c(k^{(j)}) \end{array} \right\} \tag{8}$$

where S and R represent the sets of sample indices with completely inconsistent and completely consistent voting results, respectively. l_i^{F} denotes the loss for the i-th sample in the forerunner network, and $h_j^{\mathrm{F}} = \sum_c q_c \log(q_c)$ represents the softmax prediction entropy loss for the j-th sample in the forerunner network.

3.4. Weak Augmentation Consistency Loss

To enhance the ATAN framework's learning of domain-invariant features, we construct a weak augmentation consistency loss (Xie et al., 2020), which encourages the bridge network's outputs on distorted samples to align with the outputs of the forerunner network and momentum network on the original samples. Specifically, for each sample, we applied three forms of random weak augmentation, including color jittering, horizontal flipping, and affine transformations, to generate a distorted sample. The bridge network then generates a prediction p' for this distorted sample. Finally, a symmetric cross-entropy loss is constructed between p', q and k.

$$L^{\text{wac}} = L^{\text{sce}}(p', q) + L^{\text{sce}}(p', k) \tag{9}$$

where L^{sce} is defined in the same way as in (3). Thus, the final objective function of the bridge network is formulated as shown in (10).

$$L^{B} = \frac{1}{4}(L^{\text{cst}} + L^{\text{wac}}) \tag{10}$$

where L^{cst} (Eq. (2)) means "consistency loss", L^{wac} (Eq. (9)) means "weak augmentation consistency loss".

4. Experiments

Datasets. To evaluate the effectiveness and generalizability of the proposed method, we conducted experiments on multiple datasets spanning two different types: a) Corruption datasets, including CIFAR-10-C, CIFAR-100-C, and ImageNet-C (Hendrycks and Dietterich, 2019b). These datasets are derived by applying fifteen types of corruptions to the corresponding clean test datasets of CIFAR and ImageNet, with each corruption type having 5 levels of severity. b) Natural shift datasets, including ImageNet-Sketch (1,000 categories, each containing 50 sketches) (Wang et al., 2019), and DomainNet-126 (4 domains, 126 categories) (Saito et al., 2019), where the domain distribution shifts are not caused by introduced corruptions.

Baselines. We employed four different NL-based methods as the forerunner network in ATAN—Tent (Wang et al., 2020), SAR (Niu et al., 2023), ETA (Niu et al., 2022), and ROID (Marsden et al., 2024)——to validate the effectiveness of ATAN. As a comparison, we also present the results of CoTTA (Wang et al., 2022), RoTTA (Yuan et al., 2023), RMT (Döbler et al., 2023), LAW (Park et al., 2024), OBAO (Zhu et al., 2024) and the normalization-based method BN Stats.

Implementation details. In CTTA tasks, the model is required to provide immediate predictions for online input data, without prior knowledge of domain changes, and source data is inaccessible. For datasets with shifts caused by corruptions, adaptation to 15 corruption domains is required in sequence, with the severity level of corruption set to 5 (highest) for all. For ImageNet-Sketch, only one single target domain, "Sketch", requires adaptation. For DomainNet126, after training the source model on one domain, adaptations are performed sequentially on the remaining three domains(the adaptation order is consistent with (Döbler et al., 2023)), thus requiring four pre-trained models from different

Table 1: Classification error rate (%) across different datasets in CTTA setup. In parentheses are the gains from our proposed method. * Indicates that the results are quoted from the original paper.

Method	CIFAR10-C	CIFAR100-C	ImageNet-C	ImageNet-S	DomainNet126
Source	43.5	46.4	82.0	75.9	45.3
BN Stats	20.4	35.4	68.6	73.6	41.9
CoTTA (Wang et al., 2022)	16.5	32.8	62.7	69.5	39.8
RoTTA (Yuan et al., 2023)	19.3	34.8	67.5	70.8	40.8
RMT (Döbler et al., 2023)	17.0	30.2	59.9	68.4	36.8
LAW* (Park et al., 2024)	15.7	30.9	60.1	-	-
OBAO* (Zhu et al., 2024)	15.8	29.0	59.0	-	-
Tent-A (Wang et al., 2020)	17.9	33.9	61.9	70.2	40.4
+Ours	**15.5(2.4)**	**29.3(4.6)**	**56.9(5.0)**	**67.2(3.0)**	**37.4(3.0)**
SAR (Niu et al., 2023)	18.8	31.9	62.2	68.5	40.5
+Ours	**15.5(3.3)**	**28.9(3.0)**	**57.6(4.6)**	**67.1(1.4)**	**37.4(3.1)**
ETA (Niu et al., 2022)	17.6	32.3	60.5	64.4	38.8
+Ours	**16.1(1.5)**	**29.3(3.0)**	**56.1(4.4)**	**64.1(0.3)**	**36.4(2.4)**
ROID (Marsden et al., 2024)	16.2	29.4	54.5	64.2	37.3
+Ours	**14.9(1.3)**	**27.5(1.9)**	**52.3(2.2)**	**63.4(0.8)**	**35.5(1.8)**

source domains. Gradual Test-Time Adaptation (GTTA) is a special case of CTTA (Wang et al., 2022). In GTTA setup, under each corruption domain, the severity level gradually changes from 1 to 5 and back to 1. Thus it can only be considered on corruption datasets.

When deploying ATAN, Tent used the same weight ensembling method as in ROID (Marsden et al., 2024) to enable continuous adaptation, and it was named Tent-A. The prior correction in ROID was removed, which does not affect its performance in our experimental setup. When measuring the MRF of ATAN, we used the momentum network for the second round of testing.

All pretrained model types and hyperparameters follow the settings in (Wang et al., 2022) and (Döbler et al., 2023). Specially, for CIFAR10-C, CIFAR100-C and ImageNet-C/Sketch, we apply WideResNet28 (Zagoruyko and Komodakis, 2016), ResNeXt29 (Xie et al., 2017), and ResNet50 respectively, which are from RobustBench (Croce et al., 2020) and pre-trained on corresponding clean datasets. For DomainNet126, the pre-trained ResNet50s from (Chen et al., 2022) are applied. The batch size is set to 200 for CIFAR10-C and CIFAR100-C, 64 for ImageNet-C/Sketch, and 128 for DomainNet126. The learning rate settings for each method follow the configurations used in (Döbler et al., 2023). Error rate is used as the metric. All results are averaged over 3 runs.

4.1. Results in CTTA Setup

The experimental results in CTTA setting are shown in Tab. 1. The static source model performs poorly across all test sets, highlighting the negative impact of domain shift. Simply using the test-time BN statistics (BN Stats) improves predictions. Both self-distillation-based methods (CoTTA, RoTTA and RMT) and NL-based methods (Tent-A, SAR, ETA, and ROID) further improve predictions. When using NL-based methods as the forerunner network and deploying ATAN, significant improvements in adaptation are observed across

Table 2: Classification error rate (%) across different datasets in GTTA setup. In parentheses are the gains from our proposed method.

Method	CIFAR10-C	CIFAR100-C	ImageNet-C
Source	24.7	33.6	58.4
BN Stats	13.7	29.9	48.3
CoTTA	10.9	26.3	38.8
RoTTA	11.9	33.2	55.4
RMT	9.3	26.4	39.3
LAW*	9.6	26.1	38.6
Tent-A	11.6	27.9	41.7
+Ours	**8.5(3.1)**	**23.2(4.7)**	**34.5(7.2)**
SAR	11.6	28.7	42.9
+Ours	**9.0(2.6)**	**24.7(3.0)**	**36.3(6.3)**
ETA	15.9	32.0	44.0
+Ours	**10.4(5.5)**	**25.8(6.2)**	**38.7(5.3)**
ROID	10.6	24.4	38.8
+Ours	**8.3(2.3)**	**22.5(1.9)**	**33.7(5.0)**

corruption datasets (CIFAR10-C, CIFAR100-C, ImageNet-C) and natural shift datasets (DomainNet126). This indicates that, in continual cross-domain adaptation tasks, ATAN effectively integrates knowledge from multiple domains, enhancing the model's generalization ability. In a single-domain adaptation task (ImageNet-S(ketch)), the improvement of ATAN is less noticeable, further emphasizing that ATAN's strength lies in integrating multi-domain knowledge.

4.2. Results in GTTA Setup

In Tab. 2, we report the classification error rate in GTTA setting where the severity of corruption varies progressively across the corruption datasets. Both self-distillation-based methods and NL-based methods significantly improve prediction results in this setting. When deploying ATAN with NL-based methods as the forerunner network, the adaptation performance is further enhanced. The improvement is most notable on the most challenging corruption dataset, ImageNet-C (which requires predictions across 1,000 classes). These results demonstrate that ATAN can effectively integrate multi-domain knowledge under GTTA setting, thereby greatly improving adaptation performance.

4.3. Ablation Analysis

Tab. 3 evaluates the effectiveness of different configurations within ATAN. **A** represents the configuration where Tent-A is used as the forerunner network in ATAN, L^{cst} (Eq. (2)) is employed as the loss function. **B** introduces the proposed vote-based loss adjustment mechanism (Eqs. (6) to (8)) to guide the updates of the forerunner network. **C** constructs the weak augmentation consistency loss (L^{wac}, Eq. (9)) for the bridge network. Configuration A helps the model to integrate multi-domain knowledge thus improving generalizability.

Table 3: Classification error rate (%) for different configurations. A: Siamese networks w/ L^{cst}. B: W/ vote-based loss adjustment mechanism. C: W/ L^{wac}. The table presents the results of sequentially superimposing configurations A, B, and C.

Method	CIFAR10-C	CIFAR100-C	ImageNet-C	DomainNet126
Tent-A	17.9	33.9	61.9	40.4
+ A	15.8	30.2	58.1	38.3
+ B	<u>15.6</u>	<u>29.6</u>	<u>57.7</u>	<u>38.0</u>
+ C	**15.5**	**29.3**	**56.9**	**37.2**

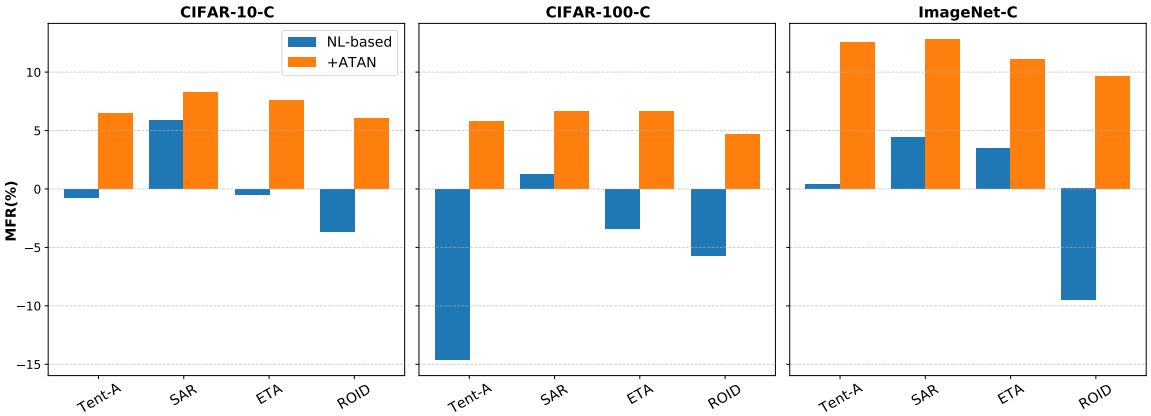

Figure 3: Mean forgetting ratio (MFR) on corruption datasets.

Configuration B helps the forerunner network to reduce error accumulation. Configuration C strengthens the model's learning of domain invariant knowledge. By sequentially adding configurations A, B, and C, ATAN significantly improves the adaptation of the source-domain model in the continually changing target domains.

4.4. Evaluation of Domain Knowledge Forgetting

To evaluate the degree of domain knowledge forgetting, we measured the mean forgetting ratio (MFR) of different methods on corruption datasets. As shown in Fig. 3, on CIFAR10-C and CIFAR100-C, three out of the four NL-based methods exhibited varying degrees of domain knowledge forgetting. Tent-A showed a forgetting ratio exceeding 10% on CIFAR100-C. On ImageNet-C, among NL-based methods, the ROID that performed best in the first round exhibited the highest degree of forgetting in the second round. SAR did not demonstrate forgetting, likely due to its gradient sharpness-aware regularization strategy. After deploying ATAN, all NL-based methods effectively leveraged the multi-domain knowledge in the momentum network to significantly mitigate forgetting and further improve predictions. In particular, the average gain on ImageNet-C is more than 10%.

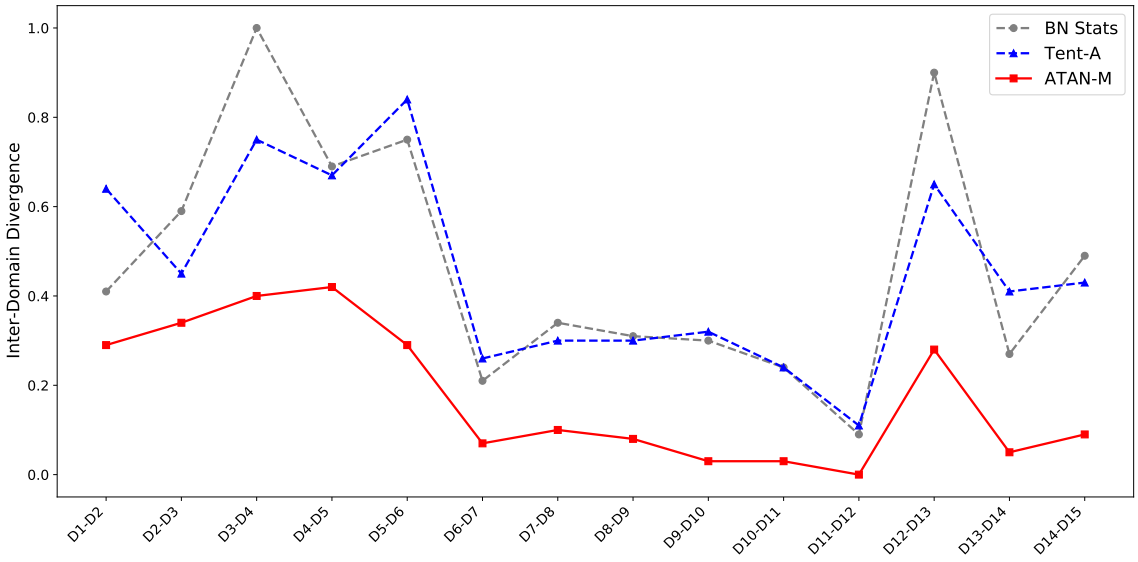

Figure 4: The inter-domain divergence of different models. BN Stats is the source model using BN statistics. ATAN use Tent-A as its forerunner network. ATAN-M indicates the momentum network from ATAN.

4.5. Inter-Domain Divergence

In order to more intuitively demonstrate the effectiveness of the proposed ATAN in integrating cross-domain knowledge, we calculated the distribution distances of the feature representations across different target domains. As in prior works (Allaway et al., 2021; Liu et al., 2024a), we use Jensen–Shannon (JS) divergence to measure inter-domain divergence between two adjacent domains. A smaller inter-domain divergence indicates that the model is less susceptible to cross-domain shifts (Ganin et al., 2016).

On CIFAR10-C, we calculate the JS divergence between the representations of adjacent domains in the order of adaptation. As shown in Fig. 4, compared to the source model using test-time batch normalization statistics (BN Stats), Tent-A does not reduce inter-domain divergence overall. However, the momentum network in ATAN significantly reduces inter-domain divergence. This indicates that ATAN effectively integrates the cross-domain knowledge, thereby exhibiting stronger consistency in feature representations across different domains.

5. Limitations

Although the proposed ATAN can significantly improve the model's generalization ability in CTTA tasks and overcome domain knowledge forgetting issues, it still has certain limitations for tasks that require lightweight and real-time performance. The limitation of ATAN lies in that it requires more forward and backward propagations. Specifically, each adaptation of ATAN typically requires 4 forward propagations, 1 global backward prop-

Table 4: Computation complexity of the proposed ATAN in combination with various NL-based methods on ImageNet-C. "SD-based" denotes the mean teacher self-distillation-based methods.

Method		Total inference time (s)	FPS	GFLOPs
NL-based	Tent-A	154.4	485.9	14.42
	SAR	279.4	268.4	25.41
	ETA	157.8	475.4	14.42
	ROID	277.9	269.9	25.41
SD-based	CoTTA	578.2	129.7	52.87
	RoTTA	660.1	113.6	60.43
ATAN	w/ Tent-A	534	140.4	48.75
	w/ SAR	654.6	114.6	59.74
	w/ ETA	529.6	141.6	48.75
	w/ ROID	640.2	117.2	58.37

agation, and 1 backward propagation of the normalization layer parameters, resulting in higher computational complexity compared to the NL-based methods. Nevertheless, compared to the standard methods based on mean teacher self-distillation, ATAN only requires backward propagation on an additional 1% of parameters (in the forerunner network). As shown in Tab. 4, the NL-based methods exhibit the lowest computational complexity. The computational complexity of the mean teacher method and ATAN are basically at the same level. Despite the above limitations, ATAN can still achieve a processing speed of over 100 frames per second on a single GPU (RTX 3090).

6. Conclusion

In this work, we discussed the issue of domain knowledge forgetting in normalization-layer-based methods under the Continual Test-Time Adaptation (CTTA) setting. To address this problem while maintaining dexterous domain adaptation ability, we proposed a method called ATAN (Anti-forgetting Test-time Adaptation Network), which consists of three Siamese networks. ATAN learns domain-specific knowledge through the forerunner network, expands the knowledge container and transfers knowledge via the bridge network, and finally integrates multi-domain knowledge through the momentum network. ATAN effectively improves model performance across various datasets and tasks, while successfully overcoming the issue of domain knowledge forgetting.

Acknowledgments

This work was supported by the National Science and Technology Major Project from the Ministry of Science and Technology, China (No. 2021ZD0201403), the National Natural Science Foundation of China (No. 62303441), the Youth Innovation Promotion Association, Chinese Academy of Sciences (No. 2021233), and the Shanghai Academic Research Leader Program (No. 22XD1424500).

References

Emily Allaway, Malavika Srikanth, and Kathleen McKeown. Adversarial learning for zero-shot stance detection on social media. *arXiv preprint arXiv:2105.06603*, 2021.

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022.

Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2090–2099, 2019.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Mario Döbler, Robert A Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7704–7714, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yulu Gan, Yan Bai, Yihang Lou, Xianzheng Ma, Renrui Zhang, Nian Shi, and Lin Luo. Decorate the newcomers: Visual domain prompt for continual test time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7595–7603, 2023.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

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

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019a.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019b.

Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

Jiaming Liu, Ran Xu, Senqiao Yang, Renrui Zhang, Qizhe Zhang, Zehui Chen, Yandong Guo, and Shanghang Zhang. Continual-mae: Adaptive distribution masked autoencoders for continual test-time adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28653–28663, 2024a.

Runze Liu, Guanghui Zhang, Dongchen Zhu, Lei Wang, Xiaolin Zhang, and Jiamao Li. Discriminative-guided diffusion-based self-supervised monocular depth estimation. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 328–342. Springer, 2024b.

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

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

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Robert A Marsden, Mario Döbler, and Bin Yang. Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2555–2565, 2024.

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

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

Junyoung Park, Jin Kim, Hyeongjun Kwon, Ilhoon Yoon, and Kwanghoon Sohn. Layer-wise auto-weighting for non-stationary test-time adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1414–1423, 2024.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8050–8058, 2019.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15922–15932, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

Zhilin Zhu, Xiaopeng Hong, Zhiheng Ma, Weijun Zhuang, Yaohui Ma, Yong Dai, and Yaowei Wang. Reshaping the online data buffering and organizing mechanism for continual test-time adaptation. In *European Conference on Computer Vision*, pages 415–433. Springer, 2024.