Open-World Test-Time Adaptation with Hierarchical Feature Aggregation and Attention Affine

Ziqiong Liu[a,*], Yushun Tang[a,*], Junyang Ji[a,b], Zhihai He[a,**]

[a]*Southern University of Science and Technology, Shenzhen, China*
[b]*Shenzhen International Graduate School, Tsinghua University, Shenzhen, China*

Abstract

Test-time adaptation (TTA) refers to adjusting the model during the testing phase to cope with changes in sample distribution and enhance the model's adaptability to new environments. In real-world scenarios, models often encounter samples from unseen (out-of-distribution, OOD) categories. Misclassifying these as known (in-distribution, ID) classes not only degrades predictive accuracy but can also impair the adaptation process, leading to further errors on subsequent ID samples. Many existing TTA methods suffer substantial performance drops under such conditions. To address this challenge, we propose a Hierarchical Ladder Network that extracts OOD features from class tokens aggregated across all Transformer layers. OOD detection performance is enhanced by combining the original model prediction with the output of the Hierarchical Ladder Network (HLN) via weighted probability fusion. To improve robustness under domain shift, we further introduce an Attention Affine Network (AAN) that adaptively refines the self-attention mechanism conditioned on the token information to better adapt to domain drift, thereby improving the classification performance of the model on datasets with domain shift. Additionally, a weighted entropy mechanism is employed to dynamically suppress the influence of low-confidence samples during adaptation. Experimental results on benchmark datasets show that our method significantly improves the performance on the most widely used classification datasets.

Keywords: Test-time Adaptation, Domain Shift, Out-of-distribution

*Equal contributions.
**Corresponding author.

1. Introduction

In real-world applications, machine learning models often encounter samples whose distributions differ significantly from the training data. Such discrepancies may arise from novel categories, sensor noise, or domain shifts [18, 28, 30, 58, 45, 54], leading to a severe degradation of predictive performance in dynamic environments [38, 50, 27]. To mitigate this issue, *Test-Time Adaptation* (TTA) has been proposed. The core idea of TTA is to dynamically update the model using unlabeled target-domain samples encountered during inference, thereby improving adaptability to new environments and distribution shifts without relying on source-domain data or target-domain labels [50, 49]. Compared to conventional domain adaptation methods, TTA is more flexible and practical, making it particularly suitable for real-world applications where annotation is costly or data privacy is restricted [3, 52, 61, 42].

Most existing TTA studies primarily focus on addressing distributional shifts caused by sensor noise or domain corruption. However, in open-world testing scenarios, models may encounter samples from unseen categories or distributions not covered by the training data [24, 61]. If these *out-of-distribution* (OOD) samples are incorrectly identified as *in-distribution* (ID) and used for adaptation, they can introduce noisy gradients that propagate through subsequent updates, ultimately leading to performance collapse [27, 55, 12]. Such misadaptation can be especially hazardous in safety-critical domains like autonomous driving and medical diagnosis [1, 19].

To address this challenge, several open-world TTA frameworks have been proposed. OSTTA [24] introduces *open-set test-time adaptation* by comparing prediction confidence before and after adaptation, filtering out samples with decreased confidence to avoid noisy updates. Despite its simplicity and effectiveness, OSTTA heavily depends on confidence-based selection and lacks explicit OOD identification, limiting its robustness in complex or long-term adaptation scenarios. OWTTT [27] adopts a prototype-based clustering strategy with distribution alignment regularization but relies strongly on source-domain data for dynamic prototype expansion. STAMP [55] dynamically updates a class-balanced memory bank with low-entropy, label-consistent samples; however, repeated augmentations and forward inferences for each sample, along with memory maintenance, introduce considerable computational overhead, reducing efficiency in real-time or resource-limited settings.

To further tackle the aforementioned limitations, we propose a novel test-time adaptation (TTA) framework that is explicitly designed to enhance

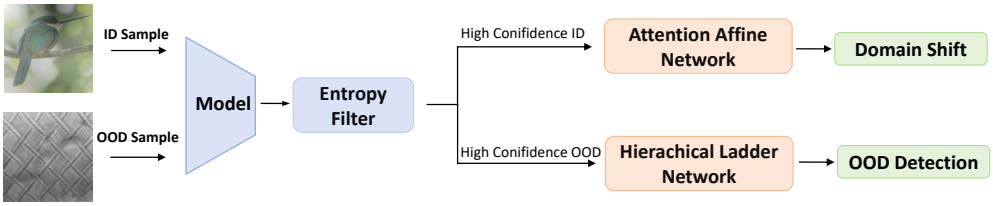

Figure 1: Diagram of our method. An entropy-based filter is applied to identify high-confidence ID and OOD samples. The ID samples are used to update the Attention Affine Network for domain shift adaptation, while the OOD samples are used to update the Hierarchical Ladder Network for improved OOD detection.

robustness against out-of-distribution (OOD) samples under open-world scenarios. Our approach incorporates a **Hierarchical Ladder Network (HLN)** that extracts OOD-relevant features from class tokens aggregated across all layers of the Transformer. Unlike prior methods that rely on shallow statistics or isolated representations, our hierarchical design captures both low-level and high-level semantic discrepancies, enabling more accurate OOD detection. We further refine predictions by fusing outputs from the base model and the HLN using a weighted probability fusion strategy, leading to more calibrated and reliable decisions under distribution shift.

Additionally, we enhance the self-attention mechanism within the Transformer architecture by introducing an **Attention Affine Network (AAN)**, which leverages token-level information to dynamically adjust attention weights. This allows the model to focus more effectively on domain-relevant features, improving its adaptability to domain drift without compromising performance on well-aligned samples. To stabilize the adaptation process, we also employ a **weighted entropy mechanism** that downweights low-confidence samples during test-time updates. This not only reduces the adverse impact of uncertain predictions but also ensures robust and consistent adaptation across diverse domain shifts. We evaluate our method on several benchmark datasets, and experimental results demonstrate its effectiveness. Our framework significantly improves classification performance and consistently outperforms existing TTA methods across various domain shift scenarios.

2. Related Work and Major Contributions

This work is related to existing research on test-time adaptation and OOD detection. In addition, this section highlights our key contributions in the context of prior work.

2.1. Test-time Adaptation

In scenarios where models must adapt to new target domains without access to source data, Test-Time Adaptation (TTA) has emerged as an effective approach for rapid model adjustment with minimal inference overhead [50]. Unlike traditional domain adaptation [13, 51] and domain generalization methods [46, 49], TTA requires no labeled target data or source samples, making it well-suited for real-world applications with scarce annotations or privacy constraints [6, 7, 40, 44, 43]. Existing TTA methods primarily fall into two categories: Batch Normalization (BN) calibration, which adapts models by updating test batch statistics and combining source and target statistics for robustness [33, 39, 56], and self-training approaches that leverage unsupervised objectives like entropy minimization and contrastive learning [50, 4] but can suffer from noisy pseudo-labels. To mitigate this, sample filtering based on entropy has been proposed [35, 36]. More recent work extends TTA to dynamic domains, label shifts, and temporal changes using specialized optimization strategies [3, 52, 61, 59]. Recent studies such as OWTTT [27] and OSTTA [24] have investigated test-time scenarios where unknown categories may be present in the data. While both works aim to enhance model generalization in open-world settings, they define distinct problem formulations: OWTTT [27] addresses the presence of entirely unknown OOD samples in the test set, whereas OSTTA [24] considers OOD samples that share similar distributional shifts with ID samples, such as noise or corruption. To better reflect the complexity of real-world conditions, our experiments are designed to encompass both scenarios. To address these challenges, a variety of strategies have been proposed. Some works adopt self-training methods based on prototype category expansion for open-world adaptation [27], while others incorporate contrastive learning to improve the separability between ID and OOD representations [41]. However, such approaches typically rely on access to source domain data as auxiliary support. An alternative line of work focuses on entropy-based optimization. For instance, OSTTA [24], UniEnt [12], and AEO [8] employ entropy minimization or regularization to distinguish OOD from ID samples. Additionally, STAMP [55] introduces a

self-weighted entropy optimization framework combined with a stable memory replay mechanism to enhance robustness.

2.2. OOD Detection

Out-of-distribution (OOD) detection is essential for ensuring the reliability and safety of deep learning models deployed in real-world applications [54]. Although these models typically perform well on in-distribution data, they often yield overconfident and unreliable predictions when presented with inputs that deviate from the training distribution, which can result in critical failures in high-risk settings [9, 27]. To mitigate this issue, numerous approaches have been proposed to detect OOD samples by analyzing model outputs without modifying the model architecture or training procedure. Common methods include maximum softmax probability (MSP) [18] and its variants such as ODIN [22, 28], energy-based techniques [29], and Mahalanobis distance-based scoring [25]. Furthermore, recent work demonstrates that incorporating additional auxiliary or outlier samples beyond the training data can enhance the model's ability to characterize the broader data distribution, thereby improving OOD detection performance and robustness against unseen inputs [37, 57].

In recent years, OOD detection research has evolved into a more comprehensive domain. Early approaches employed generative models to detect OOD samples based on reconstruction errors [15]. Follow-up work demonstrated that differences between original and reconstructed images [31], as well as dynamic features of the diffusion process trajectories—such as path change rate and curvature—can effectively identify OOD data [21, 11]. Beyond visual data, multimodal fusion techniques have also improved detection performance. For instance, MCM [32] jointly models images and text to detect OOD samples, and introducing negative prompts has been shown to enhance detection effectiveness across different scenarios [26, 34, 48].

2.3. Major Contributions

Compared to existing work, the **major contributions** of this work can be summarized as: (1) We propose a novel OOD detection method that leverages Class token information to extract discriminative features for unknown categories, effectively reducing misclassification under open-set conditions. In addition, the method maintains high ID classification accuracy by preventing excessive adaptation to uncertain samples, thus

achieving a better balance between robustness and stability during test-time adaptation. (2) We introduce an improved attention mechanism based on token information to enhance the model's adaptability to domain shift. Specifically, we design a QKV affine adaptation strategy that dynamically calibrates the Query, Key, and Value projections during test time, enabling better alignment of internal representations with the target domain and improving adaptation performance. (3) We conduct extensive experiments on several widely-used domain shift benchmarks, demonstrating that our method consistently maintains high classification accuracy on in-distribution (ID) samples, even under the interference of OOD inputs. This performance gain is primarily attributed to our effective integration of OOD-aware mechanisms into the test-time adaptation framework.

3. Method

In this section, we present our method of Hierarchical Feature Aggregation and Attention Affine for open-world test-time adaptation.

3.1. Problem Formulation and Method Overview

Suppose that a model $\mathcal{M} = f_{\theta_s}(y|X_s)$ with parameters θ_s has been successfully trained on the source data $\{X_s\}$ with labels $\{Y_s\}$. During open-world test-time adaptation, we are given the target data $\{X_t\}$ with unknown labels $\{Y_t\}$, where $\{Y_s\} \subseteq \{Y_t\}$. Our goal is to adapt the trained model to recognize the in-distribution samples and identify out-of-distribution instances to reject recognition, in an unsupervised manner during testing. Given a sequence of input sample batches $\{\mathbf{B}_1, \mathbf{B}_2, ..., \mathbf{B}_T\}$ from $\{X_t\}$, the t-th adaptation of the network model can only rely on the t-th batch of test samples $\mathbf{B}_t$.

Our method aims to dynamically adjust the Vision Transformer model during inference to mitigate performance degradation caused by domain shifts and OOD samples. The method overview is shown in Figure 2. In this work, we propose a Hierarchical Ladder Network that extracts out-of-distribution (OOD) features from class tokens aggregated across all Transformer layers. OOD detection performance is enhanced by combining the original model prediction with the output of the Hierarchical Ladder Network (HLN) via weighted probability fusion. To improve robustness under domain shift, we further introduce an Attention Affine Network (AAN) that adaptively refines the self-attention mechanism conditioned on the token information to better

adapt to domain drift, thereby improving the classification performance of the model on datasets with domain shift. Additionally, a weighted entropy mechanism is employed to dynamically suppress the influence of low-confidence samples during adaptation. In the following sections, we will further explain the proposed method in more detail.

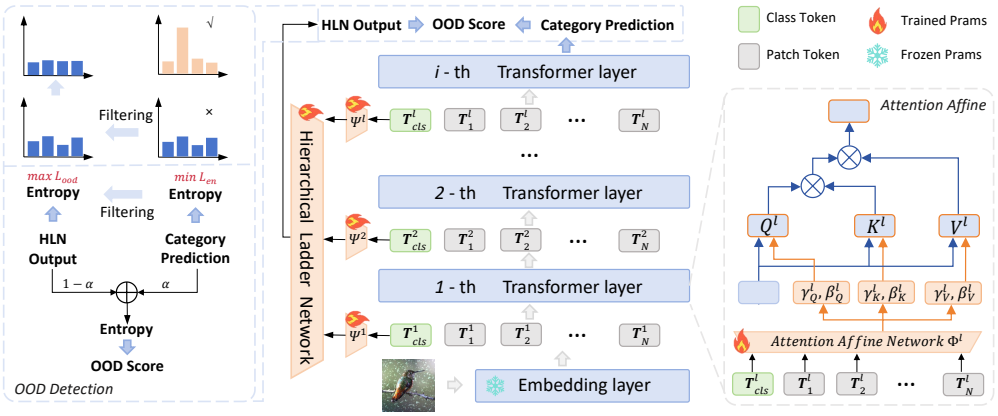

Figure 2: Overview of our proposed Hierarchical Feature Aggregation and Attention Affine.

3.2. Learning the Hierarchical Ladder Network for OOD Detection

Inspired by [55, 12], we also observe that during test-time adaptation, samples with lower entropy are more likely to belong to in-distribution (ID) classes, as they typically exhibit a sharp distribution of predicted logits. Unlike methods such as [12] and [27], which leverage class prototypes from the source domain as reference, we follow the setting in [55] and perform entropy optimization based on reliable samples. Additionally, as shown in the t-SNE visualization results in Figure 3, we observe that the separable information between OOD and ID samples is not only present in the last-layer class token, but also exists across class tokens from intermediate layers. Motivated by this observation, we incorporate the Hierarchical Ladder Network for high-confidence OOD samples to leverage multi-layer class token representations. This design enhances the separability between OOD and ID samples while maintaining classification accuracy.

Hierarchical Ladder Network. Before introducing the Hierarchical Ladder Network(HLN), we define an out-of-distribution (OOD) feature extractor Ψ^l, which operates on each layer of the transformer to extract information

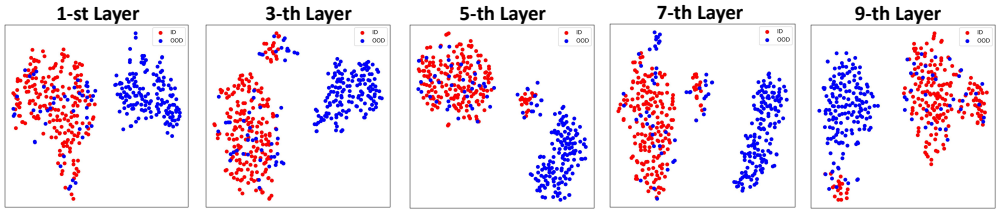

Figure 3: T-SNE visualization of the Class Tokens from different layers.

from class token. Specifically, at the l-th layer, let the class token be denoted as $\mathbf{c}_{\text{cls}}^{(l)} \in \mathbb{R}^d$, where d is the feature dimension. The OOD feature extractor takes $\mathbf{c}_{\text{cls}}^{(l)}$ as input and generates an *OOD token* $\mathbf{o}^{(l)}$ as follows:

$$\mathbf{o}^{(l)} = \Psi^l \left(\mathbf{c}_{\text{cls}}^{(l)} \right), \quad l = 1, 2, \ldots, L. \tag{1}$$

where L is the total number of transformer layers. Notably, Ψ is shared across all layers. The OOD token $\mathbf{o}^{(l)}$ is designed to carry richer out-of-distribution cues than the original class token and is used in subsequent components of the framework. The OOD token $\mathbf{o}^{(l)}$ from each layer is then forwarded to the Hierarchical Ladder Network (HLN) to generate a comprehensive OOD information token. Formally, the final OOD token is obtained as:

$$\mathbf{o}^{\text{hln}} = \text{HLN}(\{\mathbf{o}^{(l)}\}_{l=1}^{L}), \tag{2}$$

where $\text{HLN}(\cdot)$ denotes the aggregation function implemented by the Hierarchical Ladder Network. The final prediction is produced by feeding $\mathbf{o}^{\text{hln}}$ into a classifier $\mathcal{C}$, which is shared with the original class token, as follows:

$$\mathbf{p}^{\text{OOD}} = \text{softmax}(\mathcal{C}(\mathbf{o}^{\text{hln}})), \tag{3}$$

where $\mathbf{p}^{\text{OOD}}$ denotes the resulting probability distribution. Based on this distribution, the entropy of the OOD token for sample i is computed as $H_i^{\text{OOD}} = -\sum_{c=1}^{C} p_{i,c}^{\text{OOD}} \log p_{i,c}^{\text{OOD}}$, where C is the total number of classes and $p_{i,c}^{\text{OOD}}$ is the predicted probability for class c.

Loss for OOD. To encourage the model to better distinguish between in-distribution and out-of-distribution samples based on both the original model category prediction and the dedicated OOD branch, we introduce an OOD-specific loss. Specifically, we define a binary mask $\mathbf{m} \in \{0, 1\}^N$ using a predefined entropy threshold $\mathcal{H}_{\text{thr}}^{\text{OOD}}$:

$$m_i = \mathbb{I}(\mathcal{H}_i > \mathcal{H}_{\text{thr}}^{\text{OOD}}), \tag{4}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function, $\mathcal{H}_i = -\sum_{c=1}^{C} p_{i,c} \log p_{i,c}$ represents the entropy of sample i in B_t, and N is the number of samples in the current batch. As illustrated in the left part of Figure 2, the entropy $\mathcal{H}_i$ is computed based on the category prediction of sample i. This filtering process facilitates subsequent operations on the HLN outputs for OOD samples by selecting only those with relatively uniform (i.e., high-entropy) predictions. The OOD loss is then defined as the average negative entropy over the masked samples:

$$\mathcal{L}_{\text{OOD}} = -\frac{1}{\sum_{i=1}^{N} \mathbf{m}_i} \sum_{i=1}^{N} \mathbf{m}_i \cdot \mathcal{H}_i^{\text{OOD}},$$

where $\mathcal{H}_i^{\text{OOD}}$ is the entropy computed by the OOD branch. The motivation of this part is to encourage the outputs of OOD samples after the HLN network to become more uniform, which is beneficial for using OOD scores to detect such samples.

OOD Score. After the above optimization steps, we obtain the final OOD score by computing a weighted combination of the in-distribution and OOD predictions from the classifier $\mathcal{C}$:

$$\mathbf{p}^{\text{final}} = \alpha \cdot \text{softmax}(\mathcal{C}(\mathbf{c}_{\text{cls}}{}^{(L)})) + (1-\alpha) \cdot \text{softmax}(\mathcal{C}(\mathbf{o}^{\text{hln}})), \tag{5}$$

where $\alpha \in [0, 1]$ is a hyperparameter controlling the balance between in-distribution and OOD predictions, and its sensitivity is analyzed in section 4.2. Here $\mathbf{c}_{\text{cls}}{}^{(L)}$ denotes the class token at the last layer. The final OOD score is defined as the entropy of the combined output:

$$\mathcal{H}_{\text{i}}^{\text{final}} = -\sum_{c=1}^{C} p_{i,c}^{\text{final}} \log p_{i,c}^{\text{final}}, \tag{6}$$

where $p_{i,c}^{\text{final}}$ is the i-th element of $\mathbf{p}^{\text{final}}$.

3.3. Learning the Attention Affine Network for Domain Shift Correction

The self-attention mechanism is a core component of Transformer models, leveraging Query, Key, and Value (QKV) projections to compute attention weights and produce contextualized representations. However, these projections are highly sensitive to domain shifts, often resulting in degraded performance when applied to novel target domains. To address this issue, we propose the Attention Affine Network (AAN), which dynamically adjusts the

QKV projections conditioned on intermediate features at each Transformer layer. This adaptation enables more robust attention computation under domain drift and enhances generalization to unseen environments. For each attention head, AAN generates scaling/shift parameters based on patch token embeddings E:

$$[\gamma_Q^l, \beta_Q^l; \gamma_K^l, \beta_K^l; \gamma_V^l, \beta_V^l] = \Phi^l(E). \tag{7}$$

Formally, the affined Query, Key, and Value components are computed as:

$$\begin{cases} Q'^l = \gamma_Q^l \cdot Q^l + \beta_Q^l, \\ K'^l = \gamma_K^l \cdot K^l + \beta_K^l, \\ V'^l = \gamma_V^l \cdot V^l + \beta_V^l, \end{cases} \tag{8}$$

where $\gamma_Q^l, \gamma_K^l, \gamma_V^l$ are the scale factors and $\beta_Q^l, \beta_K^l, \beta_V^l$ are the shift factors for the Query, Key, and Value components in the l-th layer, respectively.

Loss for Domain Shift Since all tokens contain information and undergo similar corruptions, they may tend to capture domain shift information when assigned more probable features. To encourage similarity among patch tokens, we define the following patch-wise similarity loss:

$$\mathcal{L}_{\text{sim}} = -\frac{1}{N_{\text{patch}}} \sum_{i=1}^{N_{\text{patch}}} \sum_{j \neq i}^{N_{\text{patch}}} \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|}, \tag{9}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the patch tokens, and N_{patch} is the number of patch tokens. This loss encourages similarity by maximizing the cosine similarity between patch tokens, helping to improve the model's adaptation to domain shift.

3.4. Test-time Adaptation for Open-World

In the closed-set Test-Time Adaptation (TTA) setting, a widely adopted strategy is to update the model by minimizing the entropy of the predictive distribution, as proposed in TENT [50]. Given a mini-batch of N samples B_t at timestamp t, the entropy loss is defined as:

$$\mathcal{L}_{\text{ent}}(B_t) = -\frac{1}{N} \sum_{i=1}^{N} \mathcal{H}_i, \tag{10}$$

where $p(x) = f_{\theta_t}(x)$ denotes the softmax output of the model with parameters θ_t and the correspond predicted entropy is $\mathcal{H}_i = -\sum_{c=1}^{C} p_{i,c} \log p_{i,c}$ for sample i in B_t.

Following the entropy optimization strategy proposed in [55], we adopt a self-weighted entropy optimization approach. For a batch of N samples, the entropy loss is defined as:

$$\mathcal{L}_{\text{entropy}} = \sum_{i=1}^{N} \left(\frac{1/e^{\mathcal{H}_i}}{\sum_{j=1}^{N} 1/e^{\mathcal{H}_j}} \cdot N \right) \cdot \mathcal{H}_i, \tag{11}$$

where $\mathcal{H}_i$ denotes the entropy of the i-th sample. The total loss is composed of three terms:

$$\mathcal{L} = \mathcal{L}_{\text{entropy}} + \beta_1 \mathcal{L}_{\text{OOD}} + \beta_2 \mathcal{L}_{\text{sim}}, \tag{12}$$

where β_1 and β_2 hyperparameters. The total loss function encourages confident predictions for in-distribution data and high uncertainty for potential out-of-distribution (OOD) samples, thereby enhancing model robustness in open-world scenarios.

4. Experiments

In this section, we evaluate the effectiveness of our proposed method under the open-world test-time adaptation setting across multiple benchmark datasets. The datasets and implementation details are provided in Section 4.1, and the performance results are presented in Section 4.2.

4.1. Datasets and Implementation Details

Datasets. In the experiments, we adopt the widely used **ImageNet-C** benchmark dataset [17], which covers all categories of ImageNet-1k, including 15 types of image degradation, 5 intensity levels, and 50,000 images per level. We also use the common ImageNet-1k OOD datasets, including subsets of **Texture** [5], **Places** [60], **iNaturalist** [47] and **SUN** [53]. To evaluate the robustness of the method under the domain drift scenario, we refer to the setting of [55] and construct the corresponding perturbation datasets **Texture-C** and **Places-C**. Additionally, we evaluate on **ImageNet-R** [16], which is mainly used to test the generalization ability of the model under style transfer conditions and contains 30,000 images with diverse styles. We also include **ImageNet-A** [20], a challenging subset of ImageNet-1k consisting of naturally distributed samples that are particularly prone to classification errors.

We compare our approach with the source model (without adaptation) serving as the baseline. Our method is compared against several state-of-the-art online test-time adaptation (TTA) approaches, including TENT [50], CoTTA [52], SoTTA [14], SAR [36], OSTTA [24], UniEnt [12] and STAMP [55], with the source model (without adaptation) serving as the baseline.

Table 1: Classification Accuracy (%) for each corruption type in **ImageNet-C** at the highest severity level (Level 5) under attacks from OOD datasets (Textures, Places, iNaturalist, and SUN). The best result is shown in **bold**.

Method	Textures			Places			iNaturalist			SUN			Avg.		
	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score
Source	29.9	64.9	39.6	29.9	34.9	32.1	29.9	54.1	38.3	29.9	38.2	33.4	29.9	48.0	35.9
Tent	40.9	60.7	46.9	41.5	55.2	45.2	40.8	72.9	50.4	41.0	58.4	46.0	41.0	61.8	47.1
CoTTA	54.5	62.5	57.8	53.1	59.9	53.6	51.7	69.7	57.4	53.6	50.1	54.0	53.2	60.5	55.7
SoTTA	56.6	69.0	59.6	56.5	66.8	58.6	59.9	80.3	68.5	56.8	68.6	59.7	57.4	71.2	61.6
SAR	64.0	70.8	67.2	62.7	68.4	65.2	58.8	49.1	52.1	60.4	67.7	63.3	61.5	64.0	62.0
OSTTA	39.4	60.0	45.0	39.0	59.0	44.5	39.1	58.7	44.4	39.1	57.9	44.1	39.2	58.9	43.9
UniEnt	37.9	51.5	43.0	37.3	42.8	39.6	38.1	66.3	47.9	37.3	44.6	40.4	37.7	51.3	42.7
STAMP	59.8	72.6	64.1	62.8	65.4	64.0	63.8	88.9	73.7	62.6	68.8	65.6	62.0	73.9	66.8
Ours	**65.2**	**85.1**	**73.7**	**63.5**	**75.3**	**69.2**	**65.3**	**98.2**	**78.2**	**64.6**	**73.3**	**68.3**	**64.7**	**82.9**	**72.3**
	±0.5	±0.5	±0.6	±0.6	±0.6	±0.5	±0.3	±0.2	±0.2	±0.5	±1.4	±0.7	±0.5	±0.7	±0.5

Implementation Details.

In our experiments, all methods use the same architecture and pre-trained parameters for fair comparison. We use ViT-B/16 as backbone with a batch size of 32. The Attention Affine Network ϕ has two parts: a Token Feature Extraction Network extracting features from patch tokens, and a QKV Affine Network that combines these with the class token to independently predict shift and scale parameters for Query, Key, and Value. The QKV Affine Network is a linear layer with input d and output $6d$, generating affine transformation weights and biases. The Token Feature Extraction Network and Ψ are fully connected layers of dimension d, where Ψ, shared across layers, extracts class token information. These features feed into the Hierarchical Ladder Network, a fully connected layer with input $12d$ and output d, integrating information from all layers.

The learning rates for the Token Feature Extraction Network and the QKV Affine network ϕ are set to 0.2 and 0.0001, adding about 4.1M parameters. The OOD detection component adds 0.6M parameters, with learning rates for Ψ and the Hierarchical Ladder Network at 0.1 and 0.001, respectively. Training uses SGD optimizer. Following SAR [36], two backward passes are done per iteration: one for adapted parameters with regularization, one to

update model parameters, simulating test-time adaptation. Results report mean and std over three runs with seeds 2024, 2025, 2026. All experiments were run on an NVIDIA RTX3090 GPU. The source code will be released.

4.2. Performance Results

Table 2: Classification Accuracy (%) on **ImageNet-C** and OOD datasets (Textures-C, Places-C) under the same corruption types at severity level 5. Best results are shown in **bold**.

Method	Textures-C			Places-C			Avg.		
	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score
Source	29.9	64.9	39.6	29.9	34.9	32.1	29.9	53.9	37.0
Tent	40.1	67.7	48.9	39.1	62.5	46.6	39.6	65.1	47.7
CoTTA	52.9	67.9	57.9	54.7	65.6	59.4	53.8	66.7	58.6
SoTTA	56.2	69.8	60.7	56.5	68.1	60.7	56.4	69.0	60.7
SAR	57.6	65.3	60.9	57.7	67.6	62.0	57.6	66.4	61.4
OSTTA	39.1	55.8	44.2	37.8	56.5	43.5	38.4	56.2	43.9
UniEnt	38.2	69.4	48.1	37.8	67.8	47.0	38.0	68.6	47.6
STAMP	59.0	79.0	65.8	63.1	76.0	68.8	61.1	77.5	67.3
Ours	**64.9**	**86.0**	**73.6**	**65.1**	**77.3**	**70.6**	**65.0**	**81.7**	**72.1**
	±0.1	±0.4	±0.2	±0.6	±0.2	±0.3	±0.4	±0.3	±0.2

Table 3: Classification Accuracy (%) for each corruption in **ImageNet-R**/**A** at the highest severity (Level 5). The best result is shown in **bold**.

Source	Textures			Places			iNaturalist			SUN			Avg.		
	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score
ImageNet-R															
Source	42.7	58.2	49.2	42.7	52.5	47.1	42.7	60.5	50.1	42.7	55.7	48.3	42.7	46.7	48.7
SAR	60.8	69.3	64.8	61.5	68.8	64.9	55.4	62.8	58.8	54.9	62.1	58.3	58.1	65.8	61.7
STAMP	59.3	73.3	65.5	59.3	66.5	62.7	59.6	73.4	65.8	59.4	71.2	64.8	59.4	71.1	64.7
Ours	**63.2**	**79.8**	**70.5**	**63.3**	**75.2**	**68.8**	**62.9**	**84.2**	**72.0**	**63.0**	**82.6**	**71.5**	**63.1**	**80.5**	**70.7**
	±0.3	±0.4	±0.3	±0.6	±1.6	±1.0	±0.4	±0.4	±0.4	±0.4	±0.8	±0.6	±0.4	±0.8	±0.6
ImageNet-A															
Source	22.6	65.7	33.6	22.6	55.9	32.2	22.6	68.8	34.0	22.6	63.1	33.3	42.7	46.7	48.7
SAR	25.2	68.0	64.8	25.0	57.3	34.8	25.0	71.6	37.1	24.5	62.7	35.3	24.9	64.9	36.0
STAMP	31.0	69.0	42.8	30.8	59.4	43.7	30.8	**75.0**	43.7	30.9	**67.2**	42.3	30.9	67.6	42.3
Ours	**39.2**	**78.9**	**52.4**	**38.9**	**67.3**	**49.3**	**38.8**	70.8	**50.1**	**34.8**	64.1	**43.7**	**37.9**	**70.3**	**48.9**
	±1.5	±0.4	±1.4	±0.3	±0.1	±0.2	±0.4	±0.4	±0.4	±0.4	±0.3	±0.4	±1.2	±0.8	±1.2

Experimental Results. We systematically evaluate the performance of the proposed method on the ImageNet-C dataset and all results are rounded to one decimal place to ensure the accuracy and comparability of the data. All indicators are based on the experimental results of three groups of different random seeds, and the corresponding average values and standard

deviations under the same conditions are shown in Table 1. We follow the definition of AUC and H-score in [55]. The method is thoroughly evaluated on various OOD datasets under corruption severity level 5 on the ImageNet dataset, encompassing 15 typical corruption types such as noise, blur, weather effects, and digital disturbances. Experimental results demonstrate that in four different OOD datasets (Textures, Places, iNaturalist, and SUN), our proposed method consistently achieves superior and stable performance, with an average classification accuracy of 64.7%, outperforming the second best method by 2.7%. Furthermore, the average AUROC improves by 9%, and the overall balanced accuracy measured by the H score increases by 5.5%. We also evaluate our method on datasets following the same protocol as [12], where identical corruption types are applied to both ID and OOD samples to better reflect real-world conditions. As shown in Table 2, our method outperforms the second-best approach by 3.9%, 4.2%, and 4.8% in terms of ACC, AUC, and H-score, respectively.

To evaluate the generalization and robustness of the proposed method under real-world distribution shift conditions, we conducted comprehensive experiments on several challenging benchmark datasets. As shown in Table 3, our method achieves an H-score of 70.7% on the ImageNet-R dataset, outperforming the state-of-the-art methods SAR and STAMP by 9.0% and 4.5%, respectively, demonstrating its strong adaptability to test-time scenarios with significant appearance variations. Furthermore, on the more challenging ImageNet-A dataset, our method achieves an H-score of 48.9%, exceeding the second-best method by 6.6%.

We further extend our experiments to a larger backbone, ViT-L/16. The results, reported in Table 4, further verify the robustness and stability of our method across different Transformer backbones and diverse datasets. Detailed hyperparameter settings are provided in the Appendix.

Table 4: Classification Accuracy (%) for each corruption in **ImageNet-R** with ViT-L/16 at the highest severity (Level 5). The best result is shown in **bold**.

Source	Textures			Places			iNaturalist			SUN			Avg.		
	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score
Source	61.5	73.0	66.8	61.5	68.5	64.8	61.5	79.5	69.4	61.5	70.4	65.6	61.5	72.8	66.6
SAR	62.9	74.1	68.0	62.9	69.5	66.0	62.9	71.4	66.9	63.1	79.5	70.3	62.9	73.6	67.8
STAMP	65.4	76.5	70.5	65.5	70.5	67.9	65.4	80.3	72.1	65.4	72.1	68.6	65.4	74.8	69.8
Ours	**66.3**	**79.4**	**71.6**	**66.5**	**77.4**	**71.5**	**66.2**	**84.1**	**74.1**	**66.0**	**83.1**	**73.6**	**66.2**	**81.0**	**72.7**
	±0.5	±0.5	±0.3	±0.4	±1.1	±0.8	±0.7	±0.2	±0.6	±0.5	±1.0	±0.6	±0.5	±0.7	±0.6

As shown in Figure 4, we visualize the AUROC curves for the first and last 4000 samples. Our method consistently shows superior discriminative ability across both subsets, which further improves as adaptation proceeds. These results validate the robustness and generalizability of our method in handling diverse OOD samples under complex perturbations. The significant improvements highlight its effectiveness in addressing severe image degradation by leveraging both test-time adaptation and OOD-aware mechanisms.

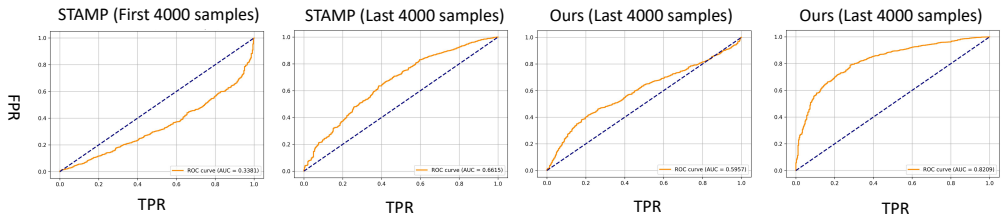

Figure 4: AUROC curve of stamp and our methods with the first 4000 samples and last 4000 samples during adpation.

Table 5: Ablation study on ImageNet-C (ID) and Textures (OOD) datasets, evaluating the individual contributions of the Attention Affine module and the Hierarchical Ladder Network.

AAN	HLN	ACC	AUC	H-score
-	-	64.3	74.3	68.9
✓	-	65.4	74.9	69.8
-	✓	64.3	83.9	72.7
✓	✓	65.0	85.0	73.5

Ablation Study. To evaluate the individual contributions of the Attention Affine Network (AAN) and Hierarchical Ladder Network (HLN), we conducted a series of controlled ablation experiments, with results summarized in Table 5. The base model, without either component, achieves 64.3% ACC, 74.3% AUC, and a 68.9% H-score. Introducing the Attention Affine module alone improves all metrics, indicating that affine calibration of Query, Key, and Value within the self-attention module enhances the model's feature representation under distribution shifts. In contrast, incorporating only the Hierarchical Ladder Network significantly boosts AUC (from 74.3% to 83.9%)

and H score (from 68.9% to 72.7%), demonstrating its effectiveness in identifying out-of-distribution samples and mitigating their impact. When both modules are combined, the model achieves the best performance: accuracy of 65.0%, AUC of 85.0% and H score of 73.5%, confirming their complementary and synergistic roles in enhancing generalization during testing.

OOD Detection Hyperparameter Analysis. In our experiments, we leverage auxiliary judgment information to generate enhanced OOD scores. Although maximizing entropy can help separate OOD samples from ID samples to some extent, our approach relies on adaptive fine-tuning with test data and does not have access to true labels, which prevents the imposition of explicit constraints on ID samples. Consequently, this limitation may indirectly increase the entropy of certain ID samples. To improve OOD detection while minimizing impact on the original classification performance, we introduce a balancing hyperparameter called the ID coefficient. Figure 5 shows the distribution of this hyperparameter, where we adopt values of 0.3 and 0.7 in our experiments.

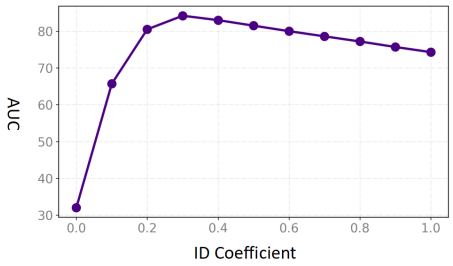

Figure 5: Comparison of the impact of different ID coefficient hyperparameters on AUC scores.

5. Conclusion

In this work, we propose a Hierarchical Ladder Network (HLN) that extracts OOD features by aggregating class tokens across all Transformer layers. OOD detection is improved by fusing the original model predictions with the HLN outputs through weighted probability fusion. Additionally, we introduce an Attention Affine Network (AAN) that adaptively refines the self-attention mechanism based on token information, enabling better adaptation to domain shifts and thereby enhancing classification performance on datasets with domain drift. Furthermore, a weighted entropy mechanism is employed to dynamically suppress the impact of low-confidence samples during adaptation. Experimental results on benchmark datasets demonstrate that our method significantly boosts performance on widely used classification tasks.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 12426312), Project 2021JC02X103, and the Center for Computational Science and Engineering at the Southern University of Science and Technology.

References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[2] Dina Bashkirova, Dan Hendrycks, Donghyun Kim, Haojin Liao, Samarth Mishra, Chandramouli Rajagopalan, Kate Saenko, Kuniaki Saito, Burhan Ul Tayyab, Piotr Teterwak, et al. Visda-2021 competition: Universal domain adaptation to improve performance on out-of-distribution data. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 66–79. PMLR, 2022.

[3] Goirik Chakrabarty, Manogna Sreenivas, and Soma Biswas. Santa: Source anchoring network and target alignment for continual test time adaptation. *Transactions on Machine Learning Research*, 2023.

[4] 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.

[5] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[6] Yuhe Ding, Jian Liang, Bo Jiang, Aihua Zheng, and Ran He. Maps: A noise-robust progressive learning approach for source-free domain adaptive keypoint detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(3):1376–1387, 2023.

[7] Yuhe Ding, Lijun Sheng, Jian Liang, Aihua Zheng, and Ran He. Proxymix: Proxy-based mixup training with label refinery for source-free domain adaptation. *Neural Networks*, 167:92–103, 2023.

[8] Hao Dong, Eleni Chatzi, and Olga Fink. Towards robust multimodal open-set test-time adaptation via adaptive entropy-aware optimization. *arXiv preprint arXiv:2501.13924*, 2025.

[9] Angelos Filos, Panagiotis Tigkas, Rowan McAllister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2020.

[10] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

[11] Ruiyuan Gao, Chenchen Zhao, Lanqing Hong, and Qiang Xu. Diffguard: Semantic mismatch-guided out-of-distribution detection using pre-trained diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1579–1589, 2023.

[12] Zhengqing Gao, Xu-Yao Zhang, and Cheng-Lin Liu. Unified entropy optimization for open-set test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23975–23984, 2024.

[13] Timnit Gebru, Judy Hoffman, and Li Fei-Fei. Fine-grained recognition in the wild: A multi-task domain adaptation approach. In *Proceedings of the IEEE international conference on computer vision*, pages 1349–1358, 2017.

[14] Taesik Gong, Yewon Kim, Taeckyung Lee, Sorn Chottananurak, and Sung-Ju Lee. Sotta: Robust test-time adaptation on noisy data streams. *Advances in Neural Information Processing Systems*, 36:14070–14093, 2023.

[15] Mark S Graham, Walter HL Pinaya, Petru-Daniel Tudosiu, Parashkev Nachev, Sebastien Ourselin, and Jorge Cardoso. Denoising diffusion models for out-of-distribution detection. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 2948–2957, 2023.

[16] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

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

[18] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[19] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[20] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021.

[21] Alvin Heng, Harold Soh, et al. Out-of-distribution detection with a single unconditional diffusion model. *Advances in Neural Information Processing Systems*, 37:43952–43974, 2024.

[22] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10951–10960, 2020.

[23] Jincen Jiang, Qianyu Zhou, Yuhang Li, Xinkui Zhao, Meili Wang, Lizhuang Ma, Jian Chang, Jian Zhang, Xuequan Lu, et al. Pcotta: Continual test-time adaptation for multi-task point cloud understanding. *Advances in Neural Information Processing Systems*, 37:96229–96253, 2024.

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

[25] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

[26] Tianqi Li, Guansong Pang, Xiao Bai, Wenjun Miao, and Jin Zheng. Learning transferable negative prompts for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17584–17594, 2024.

[27] Yushu Li, Xun Xu, Yongyi Su, and Kui Jia. On the robustness of open-world test-time training: Self-training with dynamic prototype expansion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11836–11846, 2023.

[28] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[29] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. Mood: Multi-level out-of-distribution detection. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 15313–15323, 2021.

[30] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

[31] Zhenzhen Liu, Jin Peng Zhou, Yufan Wang, and Kilian Q Weinberger. Unsupervised out-of-distribution detection with diffusion inpainting. In *International Conference on Machine Learning*, pages 22528–22538. PMLR, 2023.

[32] Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. Delving into out-of-distribution detection with vision-language representations. *Advances in neural information processing systems*, 35:35087–35102, 2022.

[33] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

[34] Jun Nie, Yonggang Zhang, Zhen Fang, Tongliang Liu, Bo Han, and Xinmei Tian. Out-of-distribution detection with negative prompts. In *The twelfth international conference on learning representations*, 2024.

[35] 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.

[36] 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.

[37] Sangha Park, Jisoo Mok, Dahuin Jung, Saehyung Lee, and Sungroh Yoon. On the powerfulness of textual outlier exposure for visual ood detection. *Advances in Neural Information Processing Systems*, 36:51675–51687, 2023.

[38] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

[39] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in neural information processing systems*, 33:11539–11551, 2020.

[40] Lijun Sheng, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Adaptguard: Defending against universal attacks for model adaptation. *arXiv preprint arXiv:2303.10594*, 2023.

[41] Houcheng Su, Mengzhu Wang, Jiao Li, Bingli Wang, Daixian Liu, and Zeheng Wang. Open-world test-time training: Self-training with contrast learning. *arXiv preprint arXiv:2409.09591*, 2024.

[42] Yushun Tang, Shuoshuo Chen, Jiyuan Jia, Yi Zhang, and Zhihai He. Domain-conditioned transformer for fully test-time adaptation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6260–6269, 2024.

[43] Yushun Tang, Shuoshuo Chen, Zhehan Kan, Yi Zhang, Qinghai Guo, and Zhihai He. Learning visual conditioning tokens to correct domain shift for fully test-time adaptation. *IEEE Transactions on Multimedia*, 2024.

[44] Yushun Tang, Shuoshuo Chen, Zhihe Lu, Xinchao Wang, and Zhihai He. Dual-path adversarial lifting for domain shift correction in online test-time adaptation. In *European Conference on Computer Vision*, pages 342–359. Springer, 2024.

[45] Yushun Tang, Ce Zhang, Heng Xu, Shuoshuo Chen, Jie Cheng, Luziwei Leng, Qinghai Guo, and Zhihai He. Neuro-modulated hebbian learning for fully test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3728–3738, 2023.

[46] Peifeng Tong, Wu Su, He Li, Jialin Ding, Zhan Haoxiang, and Song Xi Chen. Distribution free domain generalization. In *International Conference on Machine Learning*, pages 34369–34378. PMLR, 2023.

[47] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[48] Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. Clipn for zero-shot ood detection: Teaching clip to say no. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1802–1812, 2023.

[49] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8):8052–8072, 2022.

[50] Junting Wang, Mitchell Wortsman, Ning Jia, Deva Ramanan Xu, Ali Farhadi, and Kate Saenko. Tent: Fully test-time adaptation by entropy

minimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9333–9342, 2021.

[51] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[52] 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.

[53] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[54] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024.

[55] Yongcan Yu, Lijun Sheng, Ran He, and Jian Liang. Stamp: Outlier-aware test-time adaptation with stable memory replay. In *European Conference on Computer Vision*, pages 375–392. Springer, 2024.

[56] 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.

[57] Jingyang Zhang, Nathan Inkawhich, Randolph Linderman, Yiran Chen, and Hai Li. Mixture outlier exposure: Towards out-of-distribution detection in fine-grained environments. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5531–5540, 2023.

[58] Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyou Sun, Xuefeng Du, Yixuan Li, Ziwei Liu, et al. Openood v1. 5: Enhanced benchmark for out-of-distribution detection. *arXiv preprint arXiv:2306.09301*, 2023.

[59] Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. *arXiv preprint arXiv:2306.03536*, 2023.

[60] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[61] Zhi Zhou, Lan-Zhe Guo, Lin-Han Jia, Dingchu Zhang, and Yu-Feng Li. Ods: Test-time adaptation in the presence of open-world data shift. In *International Conference on Machine Learning*, pages 42574–42588. PMLR, 2023.

Appendix A. Details of Experimental Setup

Appendix A.1. Baseline and Backbone

In this study, most baseline implementations and their corresponding training strategies were adapted from the official STAMP codebase [55] to ensure faithful reproduction and fair comparison across methods. Although our overall procedure closely follows the original STAMP implementation, we adopt a Transformer-based backbone architecture and re-tune several hyperparameters—particularly the learning rate, weight decay, and adaptation-specific configurations—to better align with the characteristics of Transformer models. This design ensures both fairness and effectiveness within a unified experimental framework. Our choice is further motivated by empirical evidence suggesting that Vision Transformers achieve superior performance on large-scale datasets such as ImageNet. Experimental results corroborate this observation, showing that across most methods, the Vision Transformer backbone consistently outperforms the ResNet backbone used in STAMP in terms of both accuracy and robustness.

Appendix A.2. Hyperparameters

For all compared methods, including our proposed approach, hyperparameters are tuned consistently using the H-score metric on the Textures dataset [5], which serves as an out-of-distribution (OOD) benchmark. Specifically, we select the hyperparameter configuration that yields the best performance on this dataset and apply it directly to evaluations across all other target domains and outlier categories. This strategy ensures both consistency and fairness throughout all evaluations. The final hyperparameter settings for each method are summarized as follows:

Ours. The procedure is outlined in Algorithm 1. We set the learning rates (LR) for the main network to 0.01, 0.02, 0.05, and 0.005 for ImageNet-C, ImageNet-R, ImageNet-A, and VisDA-2021, respectively. The learning rates for the QKV affine network are set to 0.0005, 0.0005, 0.0001, and 0.0005 in the same order. For the class token feature extraction network Ψ, we use learning rates of 0.1, 0.1, 0.1, and 0.01. The learning rates for the Hierarchical Ladder Network are configured as 0.001, 0.001, 0.0001, and 0.001, respectively.

Tent. For Tent [50], we set the learning rate to LR = 0.005 for ImageNet-C. We followed the official implementation[1].

[1] `https://github.com/DequanWang/tent`

CoTTA. For CoTTA [52], we set the restoration factor to $p = 0.0005$ and the augmentation confidence threshold to $p_{\text{th}} = 0.05$. We followed the official implementation[2].
SoTTA. For SoTTA [14], we set the learning rate to LR = 0.005 and the confidence threshold to $C_0 = 0.33$. Other hyperparameters follow the original paper. We followed the official implementation[3].
SAR. For SAR [36], the learning rates are set to 0.01, 0.02, 0.05, and 0.005 for ImageNet-C, ImageNet-R, ImageNet-A, and VisDA-2021, respectively. The reset threshold is fixed at 0.1. We followed the official implementation[4].
OSTTA. For OSTTA [24], we set the learning rate to 0.001 on ImageNet-C and the loss coefficient λ_1 to 0.2. Since the official implementation is not available, we adopt the re-implementation from UniEnt[5].
UniEnt. For UniEnt [36], we set the learning rate to 0.001 on ImageNet-C. The hyperparameters λ_1 and λ_2 are set to 0.1 and 0.1, respectively. We followed the official implementation[6].
STAMP. For STAMP [55], the learning rates are set to 0.05, 0.1, 0.1, and 0.1 for ImageNet-C, ImageNet-R, ImageNet-A, and VisDA-2021, respectively. The threshold scaling factor α is fixed at 0.8. We followed the official implementation[7].

Appendix A.3. Hyperparameters for ViT-L

For this evaluation, the learning rate for the normalization layers is set to 0.0001. For the QKV Affine Network, Ψ, and the Hierarchical Ladder Network, the learning rates are configured as 0.00005, 0.0001, and 0.001, respectively. The reset constant and the learning rate for LR are set to 0.0001 and 0.2, respectively. For STAMP, the learning rate and the scaling factor α are set to 0.01 and 0.8. The batch size is fixed at 16.

Appendix B. Limitations and future work

Although we have evaluated the overall classification performance of the model under input distribution shifts and image quality degradation by testing

[2] https://github.com/qinenergy/cotta
[3] https://github.com/taeckyung/SoTTA
[4] https://github.com/mr-eggplant/SAR
[5] https://github.com/gaozhengqing/UniEnt
[6] https://github.com/gaozhengqing/UniEnt
[7] https://github.com/yuyongcan/STAMP

Algorithm 1: Unified Test-Time Adaptation with HLN and AAN under Open-World Setting

Input: Source pre-trained model f_{θ_s}; target stream $\{X_t\}$
Output: Predictions $\{\hat{\mathbf{y}}\}$ on target data

Initialize target model $f_{\theta_t} \leftarrow f_{\theta_s}$;
Initialize optimizer with tunable parameters $\tilde{\theta}$ and learning rate η;
foreach *batch* $\mathbf{B}_j \subset \{X_t\}$ **do**
    Step 1: AAN-based Feature Adaptation
    Extract patch tokens E;
    Apply Attention Affine Network (AAN) to adapt QKV projections using E;
    Step 2: HLN-based OOD Modeling
    Extract class tokens $\mathbf{c}_{\text{cls}}^{(l)}$ from each transformer layer;
    Compute OOD tokens: $\mathbf{o}^{(l)} = \Psi^l(\mathbf{c}_{\text{cls}}^{(l)})$;
    Aggregate via HLN: $\mathbf{o}^{\text{hln}} = \text{HLN}(\{\mathbf{o}^{(l)}\}_{l=1}^{L})$;
    Step 3: Forward Pass and Loss Computation
    Compute OOD predictions:
    $\mathbf{p}^{\text{final}} = \alpha \cdot \text{softmax}(\mathcal{C}(\mathbf{c}_{\text{cls}}^{(L)})) + (1-\alpha) \cdot \text{softmax}(\mathcal{C}(\mathbf{o}^{\text{hln}}))$;
    Compute entropy $\mathcal{H}_i$ for each sample;
    Compute self-weighted entropy loss $\mathcal{L}_{\text{entropy}}$;
    Compute patch similarity loss $\mathcal{L}_{\text{sim}}$ based on cosine similarity among patch tokens;
    Compute OOD loss $\mathcal{L}_{\text{OOD}}$ using masked high-entropy samples;
    Step 4: Model Update and Prediction
    Compute total loss:
    $\mathcal{L} = \mathcal{L}_{\text{entropy}} + \alpha\mathcal{L}_{\text{OOD}} + \beta\mathcal{L}_{\text{sim}}$;
    Update model parameters: $\tilde{\theta} \leftarrow \tilde{\theta} - \eta \cdot \nabla\mathcal{L}$;
    Output predictions: $\hat{\mathbf{y}} = f_{\theta_t}(\mathbf{B}_j)$;

on multiple diverse datasets—thereby simulating various scenarios where the model encounters samples different from the training set—the real-world testing environment is often more complex. It may involve a wider variety of data types, more diverse out-of-distribution (OOD) interferences, and more challenging analytical scenarios such as those encountered in 3D settings. This work does not fully address such diversity and complexity. Recent

studies [23] have explored the model's adaptability to continuously evolving target domains in 3D environments. In the future, maintaining stable model performance when confronted with novel OOD category samples in these more complex scenarios remains a practical and significant challenge.

Appendix C. Supplementary experiments

Table C.6: Classification Accuracy (%) for each corruption in **Visda-2021** at the highest severity (Level 5). The best result is shown in **bold**.

Source	Textures			Places			iNaturalist			SUN			Avg.		
	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score	ACC	AUC	H-score
Source	44.3	59.0	50.6	44.3	51.1	47.4	44.3	72.3	54.9	44.3	56.5	49.7	44.3	59.7	50.6
SAR	50.7	63.3	56.3	50.5	55.8	53.1	51.0	78.7	61.9	50.6	61.7	55.4	50.7	64.9	56.6
STAMP	57.2	66.9	61.7	57.4	66.1	61.4	57.1	83.8	67.9	57.1	59.1	58.1	57.2	69.0	62.3
Ours	**58.5**	**70.8**	**64.1**	**58.6**	**56.4**	**57.5**	**58.7**	**89.9**	**71.0**	**57.9**	**70.5**	**63.6**	**58.5**	**71.9**	**64.0**
	±0.5	±0.5	±0.3	±0.4	±1.1	±0.8	±0.7	±0.2	±0.6	±0.5	±1.0	±0.6	±0.5	±0.7	±0.6

Appendix C.1. Additional Results on VisDA-2021

To comprehensively evaluate the cross-domain adaptability of the models, we conducted experiments on the **VisDA-2021** dataset [2]. This dataset contains approximately 20,000 images from both synthetic and real domains, covering 12 object categories, and is widely regarded as a standard benchmark for domain adaptation research. To further assess out-of-distribution (OOD) robustness, we additionally incorporated **Textures** [5], **Places** [60], **iNaturalist** [47], and **SUN** [53] as potential OOD categories during testing. The corresponding results are summarized in Table C.6.

Appendix C.2. Ablation Study on the Hyperparameters

Sharpness-Aware Optimization with Custom Loss. To enhance robustness against domain shift and open-set samples during test time, we adopt the Sharpness-Aware Minimization (SAM) strategy [10], which encourages the model to converge to flat minima by optimizing over the worst-case local neighborhood. In our implementation, we define two customized loss functions used for SAM's two-step optimization:

In the first backward pass, the loss is formulated as:

$$\mathcal{L}_1(x;\Theta) = \mathcal{L}_{\text{entropy}}(x;\Theta) + \lambda_1 \cdot \mathcal{L}_{\text{OOD}}(x;\Theta) + \mathcal{L}_{\text{sim}}(x;\Theta), \tag{C.1}$$

where $\mathcal{L}_{\text{entropy}}$ denotes the entropy loss, $\mathcal{L}_{\text{OOD}}$ is the out-of-distribution (OOD) token loss, and $\mathcal{L}_{\text{sim}}$ represents the inter-class similarity loss.

Next, we compute the adversarial perturbation in parameter space via a first-order approximation:

$$\hat{\epsilon}(\Theta) = \rho \cdot \frac{\nabla_\Theta \mathcal{L}_1(x;\Theta)}{\|\nabla_\Theta \mathcal{L}_1(x;\Theta)\|_2}, \tag{C.2}$$

and perform a virtual ascent step to obtain the perturbed weights $\Theta + \hat{\epsilon}$.

In the second backward pass, the loss is defined as:

$$\mathcal{L}_2(x;\Theta+\hat{\epsilon}(\Theta)) = \mathcal{L}_{\text{entropy}}(x;\Theta+\hat{\epsilon}(\Theta)) + \lambda_2 \cdot \mathcal{L}_{\text{OOD}}(x;\Theta+\hat{\epsilon}(\Theta)), \tag{C.3}$$

which is then used to compute the final gradient for parameter updates.

This two-step optimization enhances both feature discriminability and parameter smoothness, thereby improving the model's robustness to distributional shifts. Based on comparisons of accuracy (Figure C.6), AUC (Figure C.7), and H-score (Figure C.8), we select $\lambda_1 = 0.01$ and $\lambda_2 = 0.001$ as the hyperparameters used in our experiments.

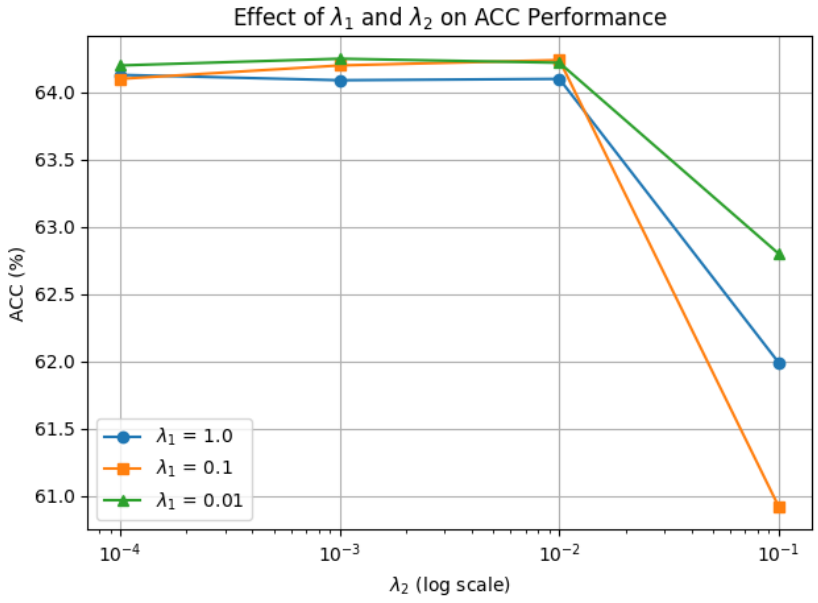

Figure C.6: The influence of regularization parameters λ_1 and λ_2 on model accuracy (ACC). The curves reflect the trend of accuracy changes with varying parameter values.

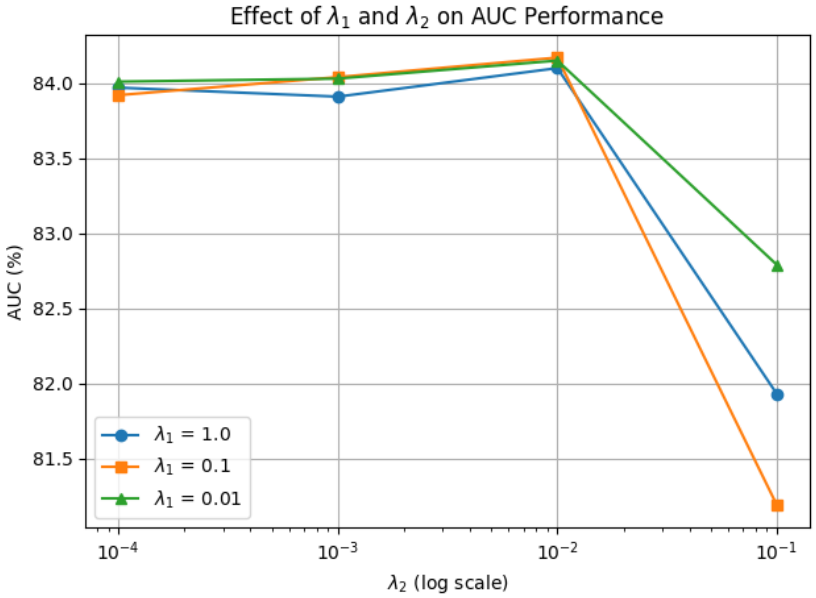

Figure C.7: The influence of regularization parameters λ_1 and λ_2 on auc score (AUC). The curves reflect the trend of accuracy changes with varying parameter values.

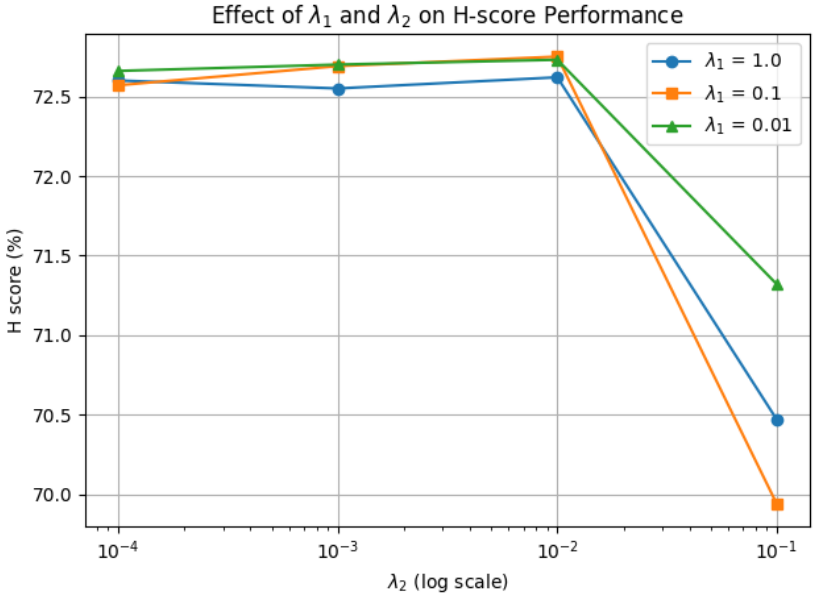

Figure C.8: The influence of regularization parameters λ_1 and λ_2 on h score (H-score). The curves reflect the trend of accuracy changes with varying parameter values.

Table C.7: Running time comparison (in minutes) on an NVIDIA RTX 3090 GPU. All methods are evaluated with ViT-B on ImageNet-C (Gaussian noise, severity level 5), using Textures as the OOD dataset (55,000 images).

Method	Source	TENT	CoTTA	SoTTA	OSTTA	UniEnt	SAR	STAMP	HLN	AAN	Ours
Time	∼3	∼6	∼79	∼22	∼8	∼11	∼10	∼75	∼10	∼25	∼26

Appendix C.3. Running Time Analysis

To provide a more comprehensive understanding of the computational efficiency of different TTA methods, we report the overall running time for completing the evaluation on ViT-B with ImageNet-C (Gaussian noise, severity level 5) and Textures as the OOD dataset (55,000 images). Table C.7 summarizes the comparison results. As shown, lightweight approaches such as Source Only and OSTTA finish within several minutes, while methods involving iterative updates or auxiliary modules (e.g., STAMP, CoTTA, UniEnt, HLN, and AAN) require significantly longer time. Our combined method (HLN + AAN) achieves a balanced trade-off between efficiency and performance, finishing in around half an hour.