Toward Efficient Test Time Adaptation With Hierarchical Distribution Alignment

Yabo Liu, Chao Huang, *Member, IEEE*, Yong Xu, *Senior Member, IEEE*, Xiaochun Cao, *Senior Member, IEEE*, and Jinghua Wang, *Member, IEEE*

***Abstract*—A model trained in a source domain often experiences a decline in effectiveness when deployed in a different target domain, primarily due to the discrepancies between the source and target domain characteristics. Test time adaptation (TTA) provides a practical solution for addressing the domain gap by adapting the models during the test phase. Existing TTA approaches mainly focus on aligning image features into a unified feature space. However, they generally only manage to achieve broad, coarse-grained alignment across domains while overlooking the more detailed, fine-grained feature clusters within each category. Furthermore, these methods are susceptible to settling at local optima because significant details can be lost when image features are abstracted into distribution parameters. To surpass these challenges, we introduce a novel approach that ensures hierarchical cross-domain alignment at three distinct levels: category-level, subcategory-level, and sample-level. Simple category-level alignment is inadequate due to the presence of various subcategories within each category, which possess distinct semantic properties identified through unsupervised clustering in our approach. Advancing further, we enhance our method by creating synthesized features from the initially extracted category-specific features, aiming for precise sample-level alignment. During our optimization process, we redefine TTA as essentially a feature matching problem, concentrating on the calculation of feature matching probabilities. Through hierarchical distribution alignment across these levels, our method maintains the semantic consistency of cross-domain image features from a broad to a detailed scale. Unlike prior test-time adaptation methods such as Tent, our method leverages source data only once after pre-training to fit feature distributions. During the testing phase, source data is completely discarded, and the model relies solely on test sample features. This design ensures privacy preservation and makes the method well-suited for privacy-sensitive applications. Our experimental evaluations on recognized datasets demonstrate that our approach significantly surpasses other established TTA methods in performance. Our code is accessible at https://github.com/yaboliudotug/HDA-TTA**

***Index Terms*—Domain adaptation, test time adaptation, transfer learning, computer vision.**

I. Introduction

THE widespread use of deep learning has greatly promoted the development of artificial intelligence. Many works have achieved success on visual tasks, such as recognition [1], [2], [3], [4], image clustering [5], [6], [7], [8], [9], anomaly detection [10], [11], [12], image segmentation [13], [14], [15], [16], and object detection [17], [18], [19], [20], [21], [22]. However, most of them assume that the training data and the testing data are drawn from the same distribution, which is often not satisfied in real-world cases. The performance of visual models decreases obviously when there are gaps between the training (or source) domain and the testing (or target) domain, which may be induced by weather change, noise effect, or blur influence.

In order to reduce the impact of domain gaps, many works propose the *test time adaptation* (TTA) method [23], [24] to learn knowledge from source domain and transfer them to the target domain. TTA assumes that we are only able to access a pre-trained source model and unlabeled target data. Different from the domain adaptation (DA) methods [25], [26], [27], [28], [29], [30], [31], which can offline train models with data from both domains, TTA methods online adapt models during the testing procedure and make predictions immediately when they receive target samples. TTA methods are also different from the test time training (TTT) methods [32], [33], which allow the modification to the structure of source models or the training losses. The TTA methods are flexible and able to deploy the pre-trained source models in new scenarios easily. Our method targets a common TTA scenario where source data can be processed offline to extract domain distribution parameters but does not need to be retained during test time. This design aligns with real-world cases, such as privacy-sensitive deployments (e.g., medical imaging) and edge computing (e.g., IoT devices), where source data cannot be stored but pre-computation is feasible.

The strategy of feature distribution alignment has shown its effectiveness in unsupervised domain adaptation [34] and test time adaptation [33], [35]. These methods assume the features

Received 11 June 2024; revised 27 April 2025 and 18 August 2025; accepted 27 September 2025. Date of publication 23 October 2025; date of current version 28 October 2025. This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant 62172285 and Grant 62301621; in part by the Guangdong Major Project of Basic and Applied Basic Research under Grant 2023B0303000010, Grant 2023A1515012110, and Grant 2025A1515011398; in part by the Shenzhen General Research Project under Grant JCYJ20241202125904007; and in part by the Shenzhen Science and Technology Program under Grant JCYJ20220818103000001, Grant 20231121172359002, and Grant 2023A008. The associate editor coordinating the review of this article and approving it for publication was Dr. Zhengming Ding. *(Corresponding author: Chao Huang.)*

Yabo Liu is with the School of Artificial Intelligence, Ocean University of China, Qingdao 266100, China (e-mail: yaboliu.ug@gmail.com).

Chao Huang and Xiaochun Cao are with the School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University, Shenzhen 518107, China (e-mail: huangch253@mail.sysu.edu.cn; caoxiaochun@mail.sysu.edu.cn).

Yong Xu and Jinghua Wang are with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: yongxu@ymail.com; wangjh2012@foxmail.com).

Digital Object Identifier 10.1109/TIP.2025.3622340

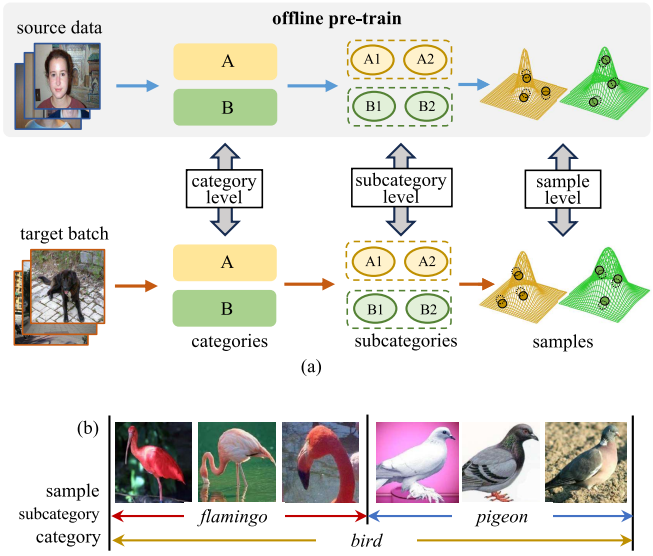


Fig. 1. (a) Is the illustration of our proposed Hierarchical Distribution Alignment method. A and B are two categories. $A1/A2$ and $B1/B2$ are subcategories within each category. We achieve hierarchical feature alignment from coarse to fine (category-subcategory-sample). (b) is the illustration of a hierarchical feature structure consisting of category, subcategory, and sample. The feature differences at the fine-grained subcategory-level not only exist in species, but also exist in other aspects *e.g.,* parts, positions, shapes, and styles.

from each category follow a certain pre-defined distribution, and then represent the features of each category with a number of parameters. For example, the work [35] models each category using a Gaussian distribution parameterized with the mean value and the covariance. These methods align the distributions by directly minimizing the KL-Divergences or covariance F-norm between the corresponding category from different domains.

However, these methods may only achieve sub-optimal performances due to several limitations. First, they simply align the coarse-grained feature distribution at the category-level and ignore the fine-grained feature distributions within each category. Because the samples associated with the same category label may have several subcategory labels based on their semantic concepts, as shown in Fig. 1 (b). In the absence of subcategory labels, the feature distribution cannot be aligned at the fine-grained level, thus affecting the efficiency of cross-domain alignment. Second, they use too few parameters to model feature distributions and ignore the characteristic of each individual feature. Note, it is impossible to fully capture the distribution of the features using a number of parameters, and the individual features can provide more and additional information about the feature distribution. The individual features can enrich randomness in the training process, and thus the model can get out saddle point and find convergence results in a larger range.

To overcome the limitations of existing TTA methods, we propose the Hierarchical Distribution Alignment method, as shown in Fig. 1 (a). To fully utilize the fine-grained feature cluster distributions within each category, we propose a subcategory-level distribution alignment strategy. Specifically, we use a k-means clustering algorithm to divide the features of each source category into several clusters, *i.e.,* subcategories. We use a subcategory classification model to fit the subcategory-wise feature clusters during the pre-training process in an offline manner. During the testing procedure, we use the pre-trained subcategory classification model to assign the target features with subcategory labels and align the features at both the category-level and the subcategory-level. In order to deal with the loss of feature diversity in existing distribution alignment methods, we propose a sample-level feature distribution alignment strategy. We generate synthesized feature samples based on the originally extracted image features with Gaussian sampling and reformulate distribution alignment as a feature matching problem. If the number of category-wise features in a target batch is insufficient, we use the fitted category-wise feature distributions to synthesize features and avoid the problem of sample imbalance in the feature matching optimization. Benefiting from the diversity of synthesized features, the efficiency and generalization of our model are improved. Note that, once the source distribution parameters are computed and the source features are extracted, the source data itself is no longer required, ensuring that test time adaptation operates independently of the source domain. This lightweight offline step is consistent with prior TTA methods such as TTAC and is well-suited for scenarios where source data must be discarded after training.

We summarize our contributions as follows:

- We propose a Hierarchical Distribution Alignment method for the Test Time Adaptation task. To the best of our knowledge, we are the first to reformulate the test time adaptation task as a feature matching problem.
- We propose to mine the subcategory feature clusters within each category and perform distribution alignment hierarchically at three levels from coarse to fine.
- We conduct extensive experiments to verify the effectiveness of our method. Experimental results demonstrate that our method achieves state-of-the-art performances.

II. Related Work

A. Unsupervised Domain Adaptation

Researchers have proposed many works to deal with the unsupervised domain adaptation problem in classification [36], [37], [38], [39], segmentation [40], [41], [42], [43], and object detection [44], [45], [46], [47]. DANN [25] is the first to utilize generative adversarial learning for domain adaptation. They propose a Gradient Reversal Layer (GRL) to reverse the gradient of model parameters. CDAN [48] uses a more fine-grained approach to learning common features within each category across domains. Reference [49] proposes to transfer domain features at the image level and use an image translation model to transfer image style across domains. Reference [50] transform the image features into graph space and use a bipartite graph approach to optimize models. Reference [13] formulate the domain adaptation as a graph matching problem and achieve good performance. ADDA [51] proposes to use different networks to extract image features and use domain discriminators to update models. To deal with the

domain adaptive object detection problem, EPM [52] pays attention to the pixel-level features to achieve fine-grained domain distribution alignment. FGRR [53] transforms the image features into the graph space. They utilize graphs to reason and aggregate information from different domains. CIGAR [54] constructs the visual graphs with image features and constructs linguistic graphs with domain labels. They use cross-modality graph reasoning between these graphs to enhance their representations.

B. Test Time Adaptation

Test Time Adaptation is a more realistic domain adaptation problem that has been widely studied [55], [56], [57], [58], [59], [60], [61]. TTT [32] is a method to adapt models during the testing time. It employs self-supervised tasks as well as supervised tasks to guide the training and models. TTT++ [33] uses a similar strategy to TTT. They use self-supervised tasks to update the models. TTT and TTT++ are classified into test time training methods. Test time adaptation assumes we cannot make any changes to the source model and the pre-training losses. In real-world applications, the setting of test time adaptation is more realistic and meaningful. When deploying a source model to an out-of-distribution target domain, test time adaptation methods can update models with streamed arrival target samples. Tent [23] is the first work to propose the fully test time adaptation. They calculate the entropy of predicted scores in the classification task. By minimizing the classification entropy they achieve the test time adaptation. The test time adaptation methods have been extended to many areas. Reference [62] use neural networks to learn the discriminative and video-specific face features to expand test time adaptation to the person re-identification.6 MemCLR [24] propose to use contrastive learning for learning domain-invariant representation in the test time adaptive object detection. STAMP [63] leverages a stable memory bank that is continuously updated by selecting low-entropy, high-quality samples. By utilizing these stored samples as References, STAMP improves its performance in open-world test-time adaptation. StickerDA [64] employs self-supervised learning to enable source models to generalize effectively to target domains. It introduces three distinct self-supervised optimization objectives, thereby enhancing adaptation performance during testing. TDA [65] integrates a memory module to store historical sample features, aimed at improving test-time adaptation for pretrained large vision-language models. During adaptation, it compares incoming test samples with stored features and adjusts its inference accordingly. TPT [66] adopts a prompt learning approach to enhance the zero-shot performance of vision-language models. By reducing inference entropy across augmented views of test samples, it increases prediction confidence and adaptation robustness. Feature alignment has been widely studied in test-time adaptation. Methods such as SHOT [67], TTAC [35], and MLFA [68] primarily focus on global and category-level alignment to address domain shifts. However, these approaches often overlook finer-grained discrepancies, such as intra-class diversity and sample-level variations. Our method builds on these works by introducing sub-category and sample-level alignment, enabling a more comprehensive and granular adaptation to domain shifts.

III. Proposed Method

Let $\mathcal{D}_s = \{x_s, y_s\}$ and $\mathcal{D}_t = \{x_t, y_t\}$ be the source domain and the target domain, where $x_{s/t}$ is an image and $y_{s/t}$ is the corresponding label. The source domain and the target domain are associated with the same label set and $y_{s/t} \in \{1, 2, \cdots, C\}$. Let $M_s = g_s \circ h_s$ be the pre-trained source model, where g_s is the backbone network and h_s is the classification head. The test time adaptation (TTA) [23], [35] aims to mine transferable knowledge from $M_s = g_s \circ h_s$ and learns a target model $M_t = g_t \circ h_t$ for the unlabeled target data $\{x_t\}$. Note, no changes can be made to the source model M_s, as its process of pre-training is independent of the testing time adaptation procedure. During the adaptation procedure, the target model M_t receives the batches of target images sequentially and produces the classification predictions immediately as each batch arrivals.

Fig. 2 shows the structure of our proposed method. We accomplish the adaptation procedure by aligning the two domains hierarchically in three levels, *i.e., the category-level*, *the subcategory-level*, and *the sample-level*. To better embed the samples from two domains, we capture the feature distributions using Gaussian distributions at both the category-level and the subcategory-level. The category-level alignment across two domains guarantees the target feature space and the source feature space have a similar semantic structure. However, the category-level alignment is not sufficient, as one category can be divided into several subcategories based on their fine-grained semantic concepts as seen in Fig. 1 (b). Thus, we propose to divide one category into a number of subcategories and achieve a fine-grained alignment at the subcategory-level. We go one step further and propose a method to achieve sample-level alignment across domains. In this way, we can take full advantage of the information encoded in each sample and avoid local optimums. To the best of our knowledge, we are the first to use feature matching to deal with the test time adaptation.

Our method is different from the existing methods in three points. First, we propose a hierarchical distribution alignment method for effective test time adaptation across domains. Second, we propose a new method to mine the subcategories within category-wise features and achieve fine-grained subcategory-level distribution alignment. Third, we propose a sample-level alignment method for the first time and such an alignment can take full advantage of the information encoded in each sample.

A. Category-Level Distribution Alignment

We use Gaussian distributions to represent the global and category-wise feature distributions following the existing work [35]. Specifically, during the offline process before adaptation, we input the source images $\{x_s\}$ into the backbone $g_s(\cdot)$ of the pre-trained source model M_s to extract their features $\{f_s\} = F_s = \{F_s^1, F_s^2, \cdots, F_s^C\}$, where $F_s^i \in \mathbb{R}^{n_i \times d}$ is the feature set of the i-th category, n_i is the number of samples belonging to the

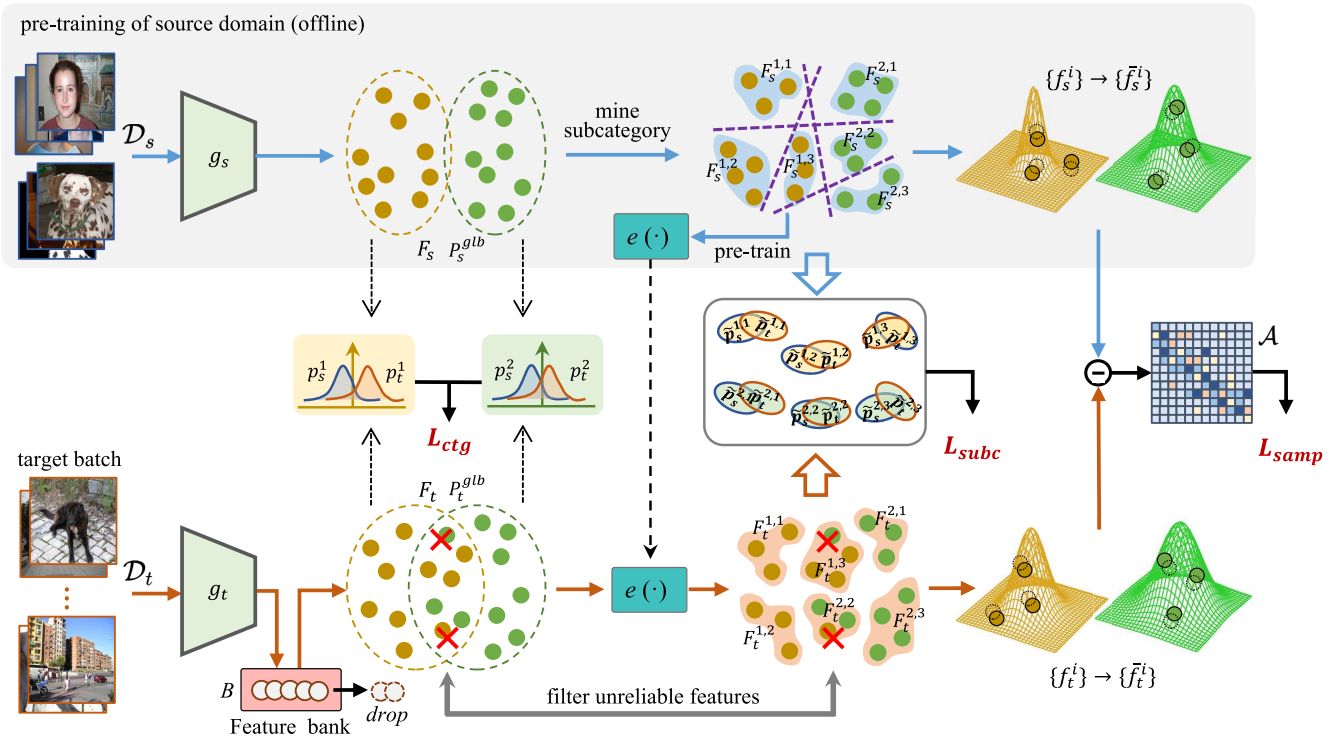


Fig. 2. Illustration of our proposed method. The backbone g_s of pre-trained model M_s extract target features F_s in an offline manner. P_s^{glb} and $p_s^{1/2}$ are Gaussian distributions that represent global and category-wise source feature distributions. K-means algorithm divides category-wise features into b subcategories which are represented by $\tilde{p}_s^{i,j}$. $e(\cdot)$ is the trained subcategory classification model. During test time adaptation, target features are stored in the feature bank B which can be used to model the online target feature distributions P_t^{glb} and $p_t^{1/2}$. $e(\cdot)$ assigns target features into subcategory clusters $\tilde{p}_t^{i,j}$. The red crosses denote filtered unreliable target features, where the pseudo subcategory labels do not match the pseudo category labels. These features are excluded from the alignment process to ensure robustness. $\bar{f}_{s/t}$ are generated novel synthesized feature samples and $\mathcal{A}$ is their matching matrix. L_{ctg}, L_{subc}, and L_{samp} are category-, subcategory-, and sample-level distribution alignment losses.

i category, and d is the feature dimension. The distribution of all source features can be described with a global Gaussian distribution:

$$P_s^{glb} = \mathcal{N}(\mu_s, \Sigma_s), \tag{1}$$

where μ_s and Σ_s are the mean value and covariance of F_s. This offline process to model the source feature distribution is computationally efficient and protects data privacy. During the test time adaptation procedure, we use the pre-trained M_s to initialize the target model $M_t = g_t \circ h_t$. As each batch of target images arrives, we extract the image features with the backbone $g_t(\cdot)$, and generate their pseudo labels with the classification head $h_t(\cdot)$. $B \in \mathbb{R}^{C \times N_{bk} \times d}$ is a first-in-first-out (FIFO) feature bank which used to store category-wise target features, where N_{bk} is the feature capacity for each category. We put the target features into B according to their pseudo labels after each batch of target images arrives sequentially. We take the target features $F_t = \{F_t^1, F_t^2, \cdots, F_t^C\}$ from feature bank and calculate their mean value and covariance (μ_t, Σ_t). Their latest global target feature distribution is shown as:

$$P_t^{glb} = \mathcal{N}(\mu_t, \Sigma_t). \tag{2}$$

In addition, we gather the category-wise source features F_s^i and model their distribution with a Gaussian distribution for each category. The overall category-wise source feature distributions can be formulated as:

$$P_s^{ctg} = \{p_s^i\}_{i=1}^C = \{\mathcal{N}(\mu_s^i, \Sigma_s^i)\}_{i=1}^C, \tag{3}$$

where μ_s^i and $\Sigma_s^i = (\sigma_s^i)^2$ are the mean value and covariance of F_s^i. Similarly, we can calculate the mean value and covariance (μ_t^i, Σ_t^i) for each category-wise target features, and model their distributions as:

$$P_t^{ctg} = \{p_t^i\}_{i=1}^C = \{\mathcal{N}(\mu_t^i, \Sigma_t^i)\}_{i=1}^C. \tag{4}$$

We use the filtering method introduced in [35] to reduce the impact of pseudo label errors. We align distributions between both category-wise and global features from different domains to embed them properly. We define the following objective as the category-level distribution alignment loss:

$$L_{ctg} = \frac{1}{C}\sum_{i=1}^{C} D_{KL}(p_s^i \| p_t^i) + D_{KL}(P_s^{glb} \| P_t^{glb}), \tag{5}$$

where $D_{KL}(\cdot\|\cdot)$ is the KL-Divergence between two distributions. The calculation of this loss relies only on the distribution parameters, *i.e.,* mean value and variance, and does not require the feature samples.

B. Subcategory-Level Distribution Alignment

Most of the existing distribution alignment methods only align domains globally or at the category-level [35], [69].

These approaches are coarse-grained because they ignore the subcategories that exist in each category. For example, within a category like "bird", features representing flamingos and pigeons can differ significantly in style, pose, or appearance. Similarly, target domain-specific noise or distortions may introduce new variations that are not adequately modeled by coarse-grained distributions. To achieve a fine-grained alignment, we propose to mine the subcategory feature clusters in each category and align them across domains. We denote this procedure as the subcategory-level distribution alignment.

1) Subcategory Discovery: In general, the classification tasks can be conducted at different levels [70]. Intuitively, features can be divided into category-level groups according to their natural category labels. In addition, these category-level features can be further divided into several subcategories based on their fine-grained semantic concepts. For example in Fig. 1 (b), both *Flamingo* and *Pigeon* have the same category label *Bird* at the task of *animal classification*, but they are visually different and should be labeled as *Flamingo* and *Pigeon* in the task of *bird classification*. In this example, we can consider *bird* as the category, and both *Flamingo* and *Pigeon* as the subcategories. Notably, the feature differences at the fine-grained subcategory level are not limited to species but also manifest in other aspects such as parts, positions, shapes, and styles [71]. To align feature distributions at the subcategory-level, we need to discover the inherent feature clusters within each category for both source and target domain features. To achieve this, we first use the k-means clustering algorithm to divide the source features F_s^i of the i-th category into b subcategory clusters $\{F_t^{i,j}\}$, where $1 \leq j \leq b$. After the unsupervised clustering process, each feature has both a category-level label y_s and a subcategory-level label $\tilde{y}_s$. To improve the reliability of subcategory-level alignment, we filter out target features whose pseudo subcategory labels do not match their pseudo category labels, as shown in Fig.2 (red crosses). This filtering mechanism reduces the impact of noisy pseudo-labels, ensuring that alignment focuses on high-confidence samples. The hierarchical feature structure of source domain features can be shown as:

$$\begin{aligned} F_s &= \{F_s^1, \cdots, F_s^C\} \\ &= \{\{F_s^{1,1}, \cdots F_s^{1,b}\}, \cdots, \{\{F_s^{C,1}, \cdots F_s^{C,b}\}\}. \end{aligned} \tag{6}$$

We then feed the source features $\{f_s\}$ into a subcategory classification model $e(\cdot)$, and obtain is prediction:

$$Softmax(e(f_s)) \in \mathbb{R}^{C \cdot b}. \tag{7}$$

We optimize $e(\cdot)$ with the Softmax loss between the predictions and subcategory labels $\tilde{y}_s$ to fit the feature clusters of source subcategories. All the subcategory discovery operations are conducted during the pre-train process in an offline manner.

2) Alignment: During the test time adaptation procedure, we feed the latest $F_s = \{f_t\}$ from the target feature bank into the pre-trained subcategory classification model $e(\cdot)$ and obtain their pseudo subcategory labels $\{\tilde{y}_t \in \mathbb{R}^{C \cdot b}\}$. We drop the target features whose subcategory labels do not match their category labels. The consistency of predicted category and subcategory labels can be used to filter out unreliable target features. According to the pseudo category and the pseudo subcategory labels of target samples, the hierarchical feature structure of target features can be expressed as:

$$\begin{aligned} F_t &= \{F_t^1, \cdots, F_t^C\} \\ &= \{\{F_t^{1,1}, \cdots F_t^{1,b}\}, \cdots, \{\{F_t^{C,1}, \cdots F_t^{C,b}\}\}. \end{aligned} \tag{8}$$

Similar to the category-level distribution alignment, we calculate the mean values $\mu_{s/t}^{i,j}$ and covariances $\Sigma_{s/t}^{i,j}$ of the j-th subcategory in the i-th category features $F_{s/t}^{i,j}$ for each domain. We use these parameters to model the Gaussian distributions of subcategory-level feature clusters as follows:

$$\tilde{P}_{s/t} = \{\{\tilde{p}_{s/t}^{i,j}\}_{j=1}^b\}_{i=1}^C = \{\{\mathcal{N}(\mu_{s/t}^{i,j}, \Sigma_{s/t}^{i,j})\}_{j=1}^b\}_{i=1}^C. \tag{9}$$

We formulate the KL-Divergences between the subcategory feature clusters across domains as the subcategory-level distribution alignment loss:

$$L_{subc} = \frac{1}{C \times b} \sum_{i=1}^{C} \sum_{j=1}^{b} D_{KL}(\tilde{p}_s^{i,j} \| \tilde{p}_t^{i,j}). \tag{10}$$

C. Sample-Level Distribution Alignment

Existing distribution alignment methods, such as those relying on Gaussian representations, abstract sample features using a limited number of parameters (e.g., mean and covariance). While computationally efficient, this abstraction smooths over the individuality of samples and fails to capture fine-grained variations, especially in cases of high intra-category variability or subtle target domain shifts. To address these challenges, we propose sample-level alignment. By generating synthesized features and reformulating the problem as feature matching, our approach retains the diversity of individual samples and prevents the loss of information critical for precise adaptation. This strategy also introduces randomness into the optimization process, which helps the model escape local optima caused by over smoothed feature representations.

To fully utilize the diversity of features for cross-domain alignment, we generate synthesized feature samples $\bar{F}_{s/t} = \{\bar{F}_{s/t}^1, \bar{F}_{s/t}^2, \cdots, \bar{F}_{s/t}^C\}$ based on the originally extracted features, where $\bar{F}_{s/t}^i = \{\bar{f}_{s/t}^i\} \in \mathbb{R}^{N_{s/t} \times d}$ are synthesized feature samples for the i-th category and $N_{s/t}$ are the numbers of generated samples. To make the synthesized features distribute similarly to the original features, we randomly select $N_{s/t}$ image features from category-wise source or target features and perform Gaussian sampling around them as follows:

$$\bar{f}_{s/t}^i = f_{s/t}^i + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 0.01). \tag{11}$$

When the numbers of category-wise original features $n_{s/t}$ are less than $N_{s/t}$, we generate $N_{s/t} - n_{s/t}$ synthesized features around the centers of $p_{s/t}^i$ in Eq. 3 and 4 for corresponding categories. In particular, we use the reparameterization trick [72] for introducing Gaussian randomness to the synthesized feature samples. It enables the Gaussian distribution parameters $\mu_{s/t}^i$ and $\sigma_{s/t}^i$ differentiable during the testing adaptation procedure and can be optimized by backpropagation. The generation process of category-wise feature samples can be detailed as:

$$\bar{f}_{s/t}^i = \mu_{s/t}^i + \sigma_{s/t}^i \times \epsilon, \quad \epsilon \sim \mathcal{N}(0, 1). \tag{12}$$

The graph matching has achieved successful performances for various visual tasks [54], [71], [73]. Inspired by these interesting approaches, we propose to reformulate the test time adaptation as a simple and efficient feature matching problem. To be specific, we first construct the feature difference matrix $W \in \mathbb{R}^{(N_s \cdot C)\times(N_t \cdot C)\times d}$ with synthesized features, where $W^{m,n} \in \mathbb{R}^d$ is the absolute difference of the m-th ($1 \le m \le (N_s \cdot C)$) source feature $\bar{F}_s^m$ in $\bar{F}_s$ and the n-th ($1 \le n \le (N_t \cdot C)$) target feature $\bar{F}_t^n$ in $\bar{F}_t$. Then we feed W into a fully connected neural network $mlp(\cdot)$ and $Sigmoid(\cdot)$ to estimate the matching matrix $\mathcal{A} \in \mathbb{R}^{(N_s \cdot C)\times(N_t \cdot C)}$ of feature pairs. The process is shown as:

$$\mathcal{A}^{m,n} = Sigmoid(mlp(abs(\bar{F}_s^m - \bar{F}_t^n))), \tag{13}$$

where $\mathcal{A}^{m,n}$ represents the probability that the m-th and n-th samples of $\bar{F}_s$ and $\bar{F}_t$ are within the same category, *i.e.,* matched feature pairs. We formulate the sample-level feature alignment loss as:

$$L_{samp} = \frac{1}{|\mathbf{Y}|} \sum_{(N_s \cdot C)\times(N_t \cdot C)} [(\mathcal{A} \odot \mathbf{Y})_{m,n} - 1]^2 + \frac{1}{|1 - \mathbf{Y}|} \sum_{(N_s \cdot C)\times(N_t \cdot C)} [\mathcal{A} \odot (1 - \mathbf{Y})]^2, \tag{14}$$

where $\mathbf{Y} \in \mathbb{R}^{(N_s \cdot C)\times(N_t \cdot C)}$ denotes the matching labels. The element $\mathbf{Y}^{m,n}$ equals 1 when the m-th and n-th samples from source and target synthesized features are matched (belonging to the same category), and 0 when they are mismatched (belonging to different categories).

By reducing domain gaps at three levels from coarse to fine, we can achieve hierarchical distribution alignment for test time adaptation. The overall adaptation loss is detailed as:

$$L = \lambda_1 L_{ctg} + \lambda_2 L_{subc} + \lambda_3 L_{samp}, \tag{15}$$

where λ_1, λ_2, and λ_3 are the weights of category-level, subcategory-level, and sample-level distribution alignment losses, respectively. Algorithm 1 shows the algorithm of our method. It consists of two parts of pre-train and test time adaptation. We take the model M_t initialized with the pre-trained source model M_s and the unlabeled target data $\{x_t\}$ as the input of our proposed method. During the testing procedure, we adapt the target model M_t as each batch of target data arrives and immediately make the predictions of the current data.

Algorithm 1 Hierarchical Distribution Alignment Algorithm

Input: Pre-trained source model M_s, N_b batches of unlabeled target domain images $\{x_t\}$, the loss weights λ_1, λ_2, and λ_3, the number of subcategories b within each category

Output: Target model M_t and the predictions of target images

Initial M_t with M_s;
Feed $\{x_s\}$ into the backbone g_s of M_s to extract $F_s = \{F_s^i\}$ $(1 \le i \le C)$;
Divide each category of F_s into b subcategory clusters $F_s^i = \{F_s^{i,j}\}$ $(1 \le i \le C, 1 \le j \le b)$;
Train $e(\cdot)$ with subcategory-wise source features;
Calculate (μ_s, Σ_s), $\{(\mu_s^i, \Sigma_s^i)\}$, and $\{(\mu_s^{i,j}, \Sigma_s^{i,j})\}$;
Formulate P_s^{glb}, $\{p_s^i\}$, and $\{\tilde{p}_s^{i,j}\}$;
Store the source global, category-wise, and subcategory-wise feature distribution parameters;
for i = 1 to N_b **do**
  Feed current batch $\{x_t\}$ into the backbone g_t of M_t to extract $F_t = \{F_t^i\}$;
  Input the target images into feature bank B and obtain latest target features;
  Divide F_t into subcategory clusters $\{F_s^{i,j}\}$ using $e(\cdot)$;
  Calculate (μ_t, Σ_t), $\{(\mu_t^i, \Sigma_t^i)\}$, and $\{(\mu_t^{i,j}, \Sigma_t^{i,j})\}$;
  Formulate P_t^{glb}, $\{p_t^i\}$, and $\{\tilde{p}_t^{i,j}\}$;
  Random sample $N_{s/t}$ features, and generate synthesized features $\bar{F}_{s/t}$;
  Calculate the category-level and subcategory-level feature distribution differences with Equations 5 and 9;
  Calculate the sample-level feature matching loss with Equation 13;
  Use $\lambda_1 L_{ctg} + \lambda_2 L_{subc} + \lambda_3 L_{samp}$ to adapt M_t;
  Output the predictions of the current batch of target images;
end for
return M_t and predictions of target images

IV. Experiments

A. Datasets

To demonstrate the effectiveness of our proposed method, we conducted experiments on the following benchmark datasets for test time adaptation.

1) CIFAR10-C: It is a corruption dataset, which consists 50k clean training images and 10k corrupted testing images with 10 categories. There are 15 types of corruptions, *i.e.,* gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transform, pixelate, and jpeg compression [74].

2) CIFAR100-C: It shares the same number of images and corruption types as CIFAR10-C. While it has a larger label space, the number of categories reaches 100 [74].

3) ImageNet-C: It is a large-scale test time adaptation dataset. It has the same corruption types as CIFAR10/100-C. It contains 1.2 million training images and 50k testing images. It has a large number of 1000 categories. The large label space brings more challenges to TTA [74].

B. Comparison Methods

We compare our method to several test time adaptation methods.

1) Source: the model only trained with images of the source domain. The source model is not fine-tuned or adapted in the target domain.

2) BN: Test time normalization updates the batch normalization statistic parameters of models according to each arrived batch during the streamed test time adaptation process [75].

3) TTT: It uses a proxy task of self-supervised learning to optimize the feature extraction ability of the model backbone [32].

4) SHOT: Source hypothesis transfer uses pseudo labels of target domain by self-supervised learning [67].

5) TENT: Test time entropy minimization updates the batch normalization statistic parameters of models by minimizing the prediction entropy of each image batch [23].

6) T3A: Test time classifier adjustment extracts the image features and calculates prototypes for each domain. It adapts domains by aligning prototypes from different domains [69].

7) TTT+: It calculate the image features of different domains. Then it aligns the feature distributions by minimizing their means and covariances [33].

8) DUA: It adapts domains by mining key samples and uses them to update the batch normalization statistic parameters of models [76].

9) EATA: It utilizes a sample selection method to identify reliable and non-redundant samples, and use them to minimize the entropy loss during testing time [77].

10) LAME: This method proposes to use the Laplacian Adjusted Maximum-likelihood Estimation approach to adapt the output of models. It can improve the quality of predictions in a training-free manner [78].

11) TTAC: It employs the feature distribution matching method to align the samples from different domains and achieves significant performance improvements [35].

12) ODS: ODS analyses the relationship between prediction error and domain shifts. Then it decouples the complex domain shifts and gradually solves them [79].

13) L-TTA: It only remodels the first layer of the backbone to adapt to the target domain. Therefore, it only requires a small involvement of the parameters [80].

14) WATT: It proposes to improve the classification performance of CLIP. The predictions are employed as pseudo labels to update the models [81].

15) ISTTA: It enhances the self-training method to improve the quality of pseudo labels and smooth the training process of model adaptation [82].

16) SAR: It improves the adaptation performance by removing the noisy samples and forcing the model to learn flat minimum parameters [83].

17) DART: During training time, it learns a prediction refinement module to analyze the class-wise confusion patterns and uses it to adapt the class distribution shifts [84].

C. Implementation Details

We use DGCNN [85] as our baseline model $M_{s/t}$ and ResNet-50 [2] as the backbone $g_{s/t}(\cdot)$ for extracting image features. We adopt the Stochastic Gradient Descent (SGD) [86] optimizer with a learning rate of 0.001 to adapt M_t. A larger batch size provides more accurate subcategory-level feature alignment by ensuring a sufficient number of samples per iteration. While excessively large batch sizes reduce computational efficiency, limiting the practical usability of the approach. For our experiments, we set the TTA batch sizes to 256 for CIFAR-10C and CIFAR-100C, and 128 for ImageNet-C. We set λ_1 and λ_3 to be 0.005 and 0.001, respectively. We set λ_2 to a relatively small value of 0.0001 to mitigate the risk of mining inaccurate subcategory distributions when the numbers of category-wise samples are limited, thereby preserving the integrity of feature alignment. The numbers N_s and N_t of generated synthesized features for each category in two domains are both set to 20. The capacity N_{bk} of target feature bank B is 2048. We set b to 3 for striking a balance between mining subcategories within category-wise features and computing memory. The initial centers of k-means algorithms are randomly initialized. We set the number of neighbors for clustering to 1. The subcategory classification model $e(\cdot)$ is a two-layer fully connected network. We adopt the Stochastic Gradient Descent optimizer with a learning rate of 0.01 and batch size of 2 for 5 epochs to train $e(\cdot)$. Our method is designed for standard test time adaptation (TTA), where each corruption type is treated as an independent target domain, and adaptation is performed separately for each corruption. The results are averaged across all corruption types to compute the final performance. All results are averaged over four independent runs to ensure statistical reliability. All experiments are conducted with NVIDIA RTX 3090 GPUs.

D. Comparison With State-Of-The-Arts

We present the comparison results of top-1 classification errors on CIFAR-10C, CIFAR-100C, and ImageNet-C in Tab. I, Tab. II, and Tab. III, respectively. The classification errors of comparison methods are inherited from [35]. The "TTAC*" in the tables represents the results implemented using the official code. All experiments are conducted at the highest severity level (5). One significant advantage of our method is its ability to use source data efficiently during the alignment process. Source data is only required once, during the initial feature alignment phase, after which it can be completely discarded. This design ensures privacy preservation while maintaining strong adaptation performance. Compared to Tent, which does not use source data, our method achieves significantly better results, particularly under larger domain shifts, demonstrating the benefits of leveraging source data during pre-alignment.

1) CIFAR-10C: Tab. I shows the comparison performances on CIFAR-10C. Our method achieves the best adaptation result with a 10.6 top-1 classification error, outperforming other existing methods. In comparison with the distribution alignment-based TTT+ and prototype-based T3A, our method achieves 3.1 and 4.8 classification error reductions. Our approach also consistently outperformed L-TTA and TTAC methods, achieving the best overall average classification error while maintaining robust results across all corruption types. This demonstrates the effectiveness of our hierarchical distribution alignment in achieving fine-grained adaptation.

2) CIFAR-100C: Tab. II lists the comparison performances on CIFAR-100C. Our method achieves a 38.3 classification error which outperforms other existing methods. Compared with batch normalization optimization-based BN and TENT, our method surpasses them by 1.2 and 2.4 classification error reductions. We also compare our method with TTT+ and achieve a 2.0 classification error reduction. The SAR and DART+SAR methods performed well on specific corruptions such as "Gaussian" and "Fog", achieving comparable results to the best-performing methods. However, on average, our method achieved the best performance of 38.3, outperforming

TABLE I

COMPARISON RESULTS ON TOP-1 CLASSIFICATION ERRORS FOR ALL CORRUPTIONS IN CIFAR-10C AT THE HIGHEST SEVERITY LEVEL. *Avg.* DENOTES THE MEAN CLASSIFICATION ERROR ACROSS ALL CORRUPTIONS OF CIFAR-10C. THE BEST RESULTS ARE MARKED IN BOLD AND THE SECOND-BEST RESULTS ARE MARKED IN UNDERLINE

Method	NOISE			BLUR				WEATHER				DIGITAL				Avg.
	Gauss	Shot	Impul	Defoc	Glass	Motn	Zoom	Snow	Frost	Fog	Birt	Contr	Elast	Pixel	Jpeg	
Source	48.7	44.0	57.0	11.8	50.8	23.4	10.8	21.9	28.3	29.4	7.0	13.3	23.4	47.9	19.5	29.2
BN	18.7	16.4	22.3	9.1	22.1	10.5	9.7	13.0	13.2	15.4	7.8	12.0	16.4	15.1	17.6	14.6
TENT	16.6	14.8	25.8	9.9	24.1	12.3	7.9	13.9	14.1	15.7	8.2	8.1	18.3	11.1	13.4	14.3
T3A	17.3	16.4	27.6	9.7	25.6	14.3	8.1	15.8	13.8	20.3	8.3	8.7	19.5	12.1	14.1	15.4
SHOT	16.0	14.9	25.0	9.1	23.3	12.6	7.5	13.8	12.6	16.9	7.6	7.8	17.8	11.4	13.2	14.0
TTT++	15.5	14.5	22.9	9.2	22.6	12.5	7.5	13.9	12.7	16.4	7.7	7.9	17.6	11.5	13.0	13.7
TTT	45.6	41.8	50.0	21.8	46.1	23.0	23.9	29.9	30.0	25.1	12.2	23.9	22.6	47.2	27.2	31.4
DUA	15.4	13.4	17.3	8.0	18.0	9.1	7.7	10.8	10.8	12.1	6.6	10.9	**13.6**	13.0	14.3	12.1
EATA	51.1	51.8	58.0	34.6	56.6	40.2	42.7	44.9	47.0	44.0	40.5	38.5	48.7	44.3	49.1	46.1
LAME	19.4	18.2	29.9	8.5	19.9	12.0	7.3	13.7	11.6	16.4	8.7	10.5	16.0	9.5	19.7	14.7
TTAC*	13.2	11.9	17.1	7.8	18.4	9.1	6.5	10.1	10.6	11.1	**6.1**	9.0	14.7	9.5	10.8	11.1
ODS	32.5	34.2	28.1	11.3	43.6	9.5	11.9	13.8	13.1	16.0	12.6	8.8	20.6	15.6	17.9	19.3
L-TTA	22.4	18.8	28.3	7.8	30.4	9.9	**6.5**	**8.6**	**9.3**	**9.2**	4.2	5.6	17.3	11.6	22.1	14.1
WATT	36.1	34.8	41.4	21.1	34.9	22.2	20.7	20.3	19.5	21.5	12.9	18.8	27.4	28.9	32.7	26.2
SAR	17.5	16.2	25.4	8.3	18.4	11.0	7.6	12.1	11.0	11.9	8.5	9.2	15.4	9.3	16.2	13.2
DART+SAR	17.5	16.3	27.2	8.1	**18.3**	10.8	7.6	12.1	10.9	12.0	8.4	9.1	15.5	**9.2**	16.4	13.3
Ours	**10.8** ±1.6	**11.5** ±0.1	**16.4** ±0.8	**7.6** ±1.1	18.7 ±2.1	**8.8** ±0.7	6.6 ±1.5	9.8 ±2.1	10.1 ±0.8	10.4 ±2.3	6.3 ±3.2	**7.5** ±0.5	14.1 ±3.2	9.4 ±2.7	**10.7** ±1.3	**10.6** ±1.6

TABLE II

COMPARISON RESULTS ON TOP-1 CLASSIFICATION ERRORS FOR ALL CORRUPTIONS IN CIFAR-100C AT THE HIGHEST SEVERITY LEVEL. *Avg.* DENOTES THE MEAN CLASSIFICATION ERROR ACROSS ALL CORRUPTIONS OF CIFAR-100C. THE BEST RESULTS ARE MARKED IN BOLD AND THE SECOND-BEST RESULTS ARE MARKED IN UNDERLINE

Method	NOISE			BLUR				WEATHER				DIGITAL				Avg.
	Gauss	Shot	Impul	Defoc	Glass	Motn	Zoom	Snow	Frost	Fog	Birt	Contr	Elast	Pixel	Jpeg	
Source	80.8	77.8	87.8	39.6	82.3	54.2	38.4	54.6	60.2	68.1	**28.8**	50.9	59.5	72.3	50.0	60.3
BN	44.7	44.2	**47.4**	32.4	46.4	**32.9**	33.0	39.0	38.4	45.3	30.2	36.6	40.6	37.2	44.2	39.5
TENT	45.6	43.4	56.2	32.5	52.3	38.4	31.1	42.9	41.4	44.9	30.5	31.5	45.8	35.6	38.9	40.7
T3A	46.6	45.3	58.9	33.6	55.5	40.3	31.4	46.1	42.0	53.1	31.7	32.6	47.6	37.2	38.9	42.7
SHOT	43.4	42.2	53.8	31.3	51.0	36.1	29.5	41.0	39.3	45.1	29.4	30.5	43.4	34.5	36.1	39.1
TTT++	43.9	42.3	54.1	33.0	52.1	38.1	30.6	43.0	40.2	47.7	30.8	31.5	45.0	35.4	37.3	40.3
EATA	44.8	41.9	52.6	33.0	51.1	37.8	30.3	43.0	40.1	45.1	30.1	31.8	45.2	35.2	37.4	40.0
LAME	61.7	60.8	68.4	35.5	59.2	37.6	35.6	47.3	47.0	43.6	32.2	38.4	46.8	43.6	57.3	47.7
TTAC*	**41.5**	**40.2**	51.9	31.1	56.2	34.7	**28.3**	39.9	38.4	43.2	30.0	39.6	43.9	33.8	35.9	39.2
ODS+NOTE	55.8	50.9	59.1	35.5	54.0	34.1	34.6	39.9	38.4	40.7	29.2	**30.5**	42.2	41.5	48.9	42.3
WATT	68.0	65.7	69.7	47.1	67.9	49.5	44.7	47.3	46.3	48.6	36.5	47.3	59.1	59.1	60.5	54.5
SAR	46.1	45.2	54.0	30.9	45.1	34.7	29.3	39.7	37.6	39.8	32.1	33.1	41.4	32.8	43.9	39.0
DART+SAR	46.1	45.2	54.9	30.9	**45.1**	34.7	29.3	39.7	**37.6**	**39.8**	32.1	33.1	**41.4**	**32.8**	43.9	39.1
Ours	41.6 ±4.2	40.3 ±2.5	51.1 ±1.1	**30.7** ±4.8	50.1 ±2.9	34.8 ±2.2	28.5 ±2.4	**39.0** ±1.3	38.3 ±3.7	43.2 ±1.5	28.9 ±3.4	34.9 ±3.7	43.0 ±1.4	34.2 ±4.0	**35.5** ±4.3	**38.3** ±3.0

TABLE III

COMPARISON RESULTS ON TOP-1 CLASSIFICATION ERRORS FOR ALL CORRUPTIONS IN IMAGENET-C AT THE HIGHEST SEVERITY LEVEL. *Avg.* DENOTES THE MEAN CLASSIFICATION ERROR ACROSS ALL CORRUPTIONS OF IMAGENET-C. THE BEST RESULTS ARE MARKED IN BOLD AND THE SECOND-BEST RESULTS ARE MARKED IN UNDERLINE

Method	NOISE			BLUR				WEATHER				DIGITAL				Avg.
	Gauss	Shot	Impul	Defoc	Glass	Motn	Zoom	Snow	Frost	Fog	Birt	Contr	Elast	Pixel	Jpeg	
Source	97.3	96.7	97.8	82.2	90.5	83.6	74.3	82.2	76.8	64.8	38.8	89.6	87.1	80.4	68.3	80.7
BN	**60.2**	**60.7**	**59.8**	76.6	**68.7**	67.4	64.2	64.6	66.2	74.7	57.0	88.8	55.8	53.0	52.3	64.7
TENT	84.9	84.2	81.4	75.7	85.0	49.5	43.5	49.2	64.3	35.4	31.4	40.3	42.0	39.8	46.8	56.9
SHOT	76.3	70.6	74.2	**62.0**	71.9	**48.0**	**40.7**	46.1	54.2	**34.7**	30.7	**37.7**	41.3	**38.9**	46.5	51.6
TTT	75.5	76.8	81.9	89.6	82.8	79.1	71.3	83.6	81.0	98.3	59.0	99.0	54.7	53.2	49.6	75.7
DUA	71.9	72.6	72.4	90.2	80.8	83.1	74.7	76.4	77.9	87.3	62.6	99.3	60.8	58.4	52.6	74.7
TTAC*	66.8	62.6	64.8	69.7	71.2	52.9	43.2	44.5	50.3	35.6	**30.4**	39.6	**39.7**	39.9	**45.6**	**50.5**
ODS	70.3	67.3	67.8	71.3	72.0	55.6	48.6	50.6	59.7	40.9	31.7	82.7	43.3	40.0	46.0	56.5
L-TTA	79.4	78.8	82.2	75.8	81.1	72.4	63.5	68.8	58.5	53.5	33.4	73.1	57.0	41.4	54.5	64.9
ISTTA	72.9	70.8	73.1	80.7	79.7	69.6	57.4	59.8	63.1	50.0	39.3	83.9	51.8	48.5	50.8	63.4
SAR	73.5	74.2	72.8	74.6	75.3	62.8	53.4	56.5	60.2	44.7	33.3	67.4	48.6	43.9	50.5	59.4
DART+SAR	73.5	74.3	72.8	74.6	75.3	62.8	53.4	56.5	60.2	44.7	33.3	67.4	48.6	43.9	50.6	59.5
Ours	67.1 ±3.5	62.7 ±3.9	65.1 ±5.0	71.7 ±4.1	70.6 ±4.7	53.5 ±4.3	43.2 ±2.6	**44.3** ±3.1	**50.1** ±1.5	35.9 ±4.7	30.5 ±2.4	40.6 ±2.9	40.1 ±2.1	40.5 ±2.3	45.8 ±1.6	50.8 ±3.4

TABLE IV

IMPACT OF A DIFFERENT NUMBER OF SUBCATEGORIES b WITHIN EACH CATEGORY. W/O DENOTES $b = 1$, *i.e.,* DO NOT USE SUBCATEGORY-LEVEL DISTRIBUTION ALIGNMENT LOSS. WE PRESENT THE CLASSIFICATION ERRORS OF FOUR CORRUPTIONS OF *Gaussian*, *Motion*, *Snow*, AND *Contrast*. *Average* DENOTES THE MEAN CLASSIFICATION ERROR ACROSS ALL CORRUPTIONS

Corruptions	Gaussian	Motion	Snow	Contrast	Average
w/o	13.51	8.96	9.83	8.09	10.85
2	12.89	9.13	**9.70**	7.65	10.73
3	**10.80**	**8.77**	9.79	**7.48**	**10.57**
4	13.14	9.00	10.01	7.64	10.71

both SAR (39.0) and DART+SAR (39.1). This indicates that while SAR-based methods are effective for certain scenarios, our hierarchical alignment approach offers superior adaptability across diverse corruption types.

3) ImageNet-C: As shown in Tab. III, we present the comparison results on ImageNet-C. Our method shows the best performance on the large-scale test time adaptation dataset with large label space. In comparison with the batch normalization optimization-based BN, TENT, and DUA, our method reduces the classification error by a margin of 13.9, 6.1, and 23.9, respectively. Our method surpasses the self-supervised-based SHOT and TTT by 0.8 and 24.9 classification error reductions. SHOT and TTAC delivered relatively strong results for ImageNet-C, particularly on weather-related corruptions (e.g., "Snow" and "Fog"), where their detailed adaptation techniques excelled. Notably, TTAC achieved the best overall average classification error, narrowly outperforming our method. However, our approach showed more consistent performance across all categories, particularly excelling in digital corruptions such as "JPEG" and "Contrast". This consistency underscores the strength of our method for large-scale datasets with diverse corruption patterns.

E. Ablation Studies

In this section, we conduct ablation studies to verify the effectiveness of each module in our method. The experiments are studied on the CIFAR-10C corruption dataset at the highest severity level (5).

1) Ablation on Subcategory-Level Feature Alignment: We compare the adaptation results with a different number of subcategories within each category in our method. Tab. IV shows the classification errors of four typical corruptions and the average of all 15 corruptions. The subcategory-level adaptation method can reduce a margin of 0.28 average classification error at most. We also investigate the influence of the number of subcategories b within each category. As b increases from 1 to 3, the classification error gradually decreases. When b is larger than 3, the classification error increases because the sample number of each subcategory is too small. To balance the performance and efficiency of our method, we set b to 3 for all experiments.

2) Ablation on Distribution Alignment Losses: We investigate the effectiveness of each component in our proposed alignment loss. As shown in Tab. V, all loss components in our method can reduce the classification errors compared with the baseline (only L_{ctg}) method. When L_{subc} and L_{samp} are added to L_{ctg}, the classification errors decrease by 0.17 and 0.22, respectively. When all three components (L_{ctg}, L_{subc}, and L_{samp}) are utilized, the system achieves an optimal performance with a classification error of 10.6. Compared to the baseline method, incorporating the complete hierarchical distribution alignment loss results in a reduction of the classification error by 0.5. This study indicates the effectiveness of each proposed module.

TABLE V

COMPARISON RESULTS ON DIFFERENT COMPONENTS OF HIERARCHICAL DISTRIBUTION ALIGNMENT LOSS. *Average* DENOTES THE MEAN CLASSIFICATION ERROR ACROSS ALL CORRUPTIONS

L_{ctg}	L_{subc}	L_{sample}	Average
✔	-	-	11.07
✔	✔	-	10.90
✔	-	✔	10.85
✔	✔	✔	**10.57**

F. Qualitative Results

Fig. 3 presents the visualization results for sample-level feature matching of CIFAR-10C. The rectangle on the left denotes the matching label $\mathbf{Y}$. The white pixel $\mathbf{Y}^{m,n}$ located at (m, n) indicate that the feature pair are matched, *i.e.,* belong to the same category. The rectangle on the right denotes the matching matrix $\mathcal{A}$ of feature pairs. The element $\mathcal{A}^{m,n} \in [0, 1]$ in the matrix indicates the predicted probability that m-th and n-th samples of $\bar{F}_s$ and $\bar{F}_t$ are matched. The redder the color of an element, the closer its value is to 1. Conversely, the darker the color, the closer the value is to 0. At the beginning of the testing, the model could not distinguish the matched feature pairs correctly and predicted vague matching probabilities of about 0.5 for all feature pairs. As the batches of target samples arrive in sequence, the number and confidence of correctly matched feature pairs increase. The visualization demonstrates that our feature matching method can prompt sample-level feature alignment across domains.

G. Alternative Strategy for Sample-Level Alignment

We propose the sample-level feature matching method to achieve fine-grained adaptation across domains. In particular, we generate synthesized features according to the originally extracted features to ensure that the synthesized samples are similar to the real ones. This operation complies with the setting of test time adaptation (TTA) that does not access

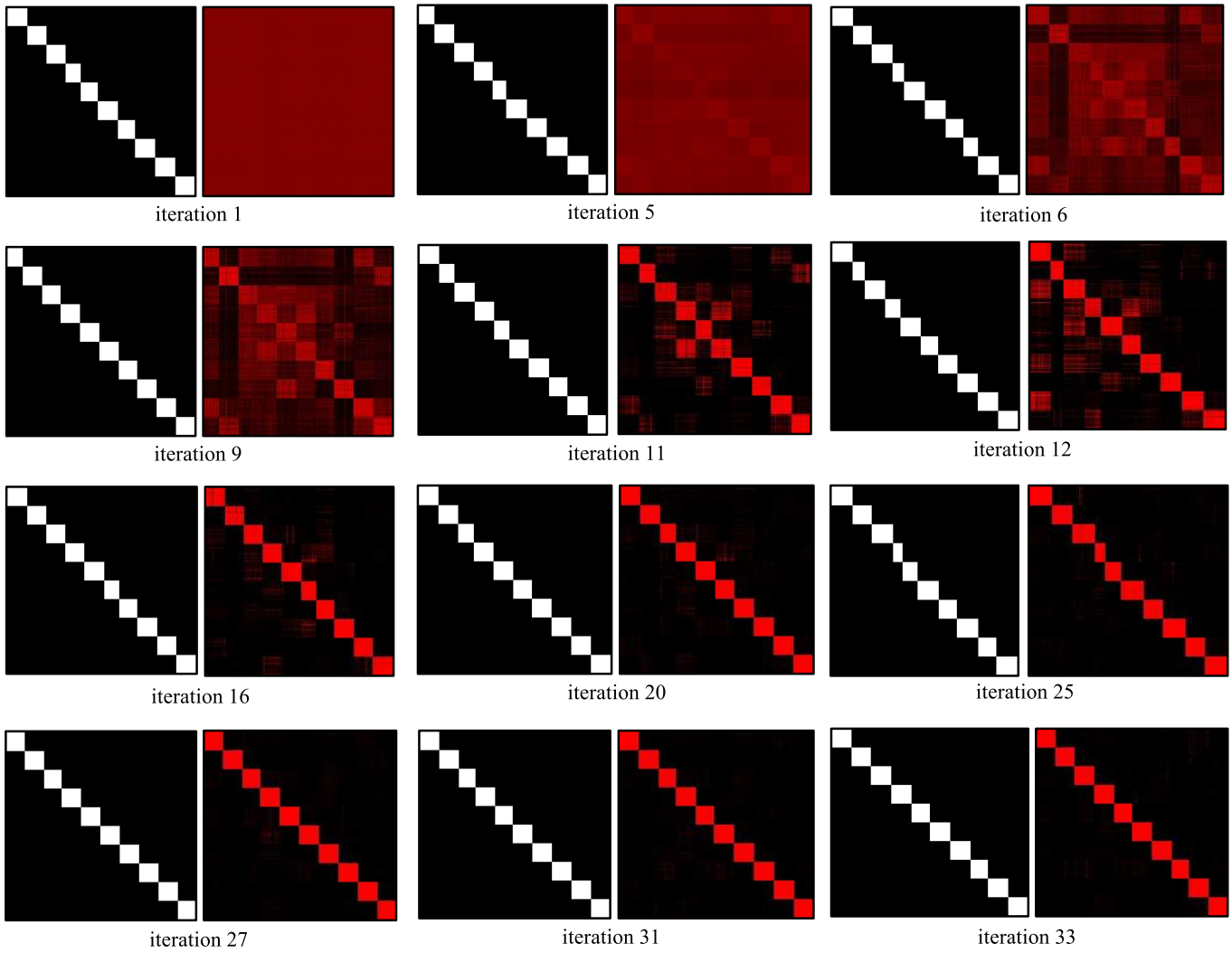


Fig. 3. Visualization of the feature matching during the testing procedure from 1 to 33 iterations. The white areas of left rectangles indicate that the feature pairs at the corresponding location is matched, *i.e.,* belongs to the same category. The left rectangles are matching matrixes $\mathcal{A}$ at different testing iterations. The active red element $\mathcal{A}^{m,n}$ in the matrix indicates that the predicted matching probability that m-th and n-th samples of $\bar{F}_s$ and $\bar{F}_t$ are matched.

TABLE VI

COMPARISON RESULTS ON SAMPLE-LEVEL FEATURE ALIGNMENT. *Distribution* AND *Feature* DENOTE GENERATE SYNTHESIZED FEATURES WITH CATEGORY-WISE DISTRIBUTIONS AND ORIGINAL FEATURES, RESPECTIVELY

Corruptions	CIFAR-10C		CIFAR-100C		Imagenet-C	
	Distribution	Feature	Distribution	Feature	Distribution	Feature
Gauss	12.89	10.80	41.87	41.62	67.10	67.11
Shot	11.70	11.51	40.54	40.29	62.82	62.70
Impul	16.27	16.41	51.09	51.14	65.17	65.14
Defoc	7.56	7.63	30.84	30.70	71.95	71.70
Glass	18.64	18.66	50.73	50.10	70.75	70.60
Motn	9.13	8.77	35.00	34.81	53.60	53.51
Zoom	6.39	6.61	28.74	28.53	43.14	43.24
Snow	9.70	9.79	39.04	39.04	44.17	44.32
Frost	10.16	10.14	38.16	38.26	49.99	50.14
Fog	9.84	10.35	43.64	43.20	35.70	35.93
Birt	6.15	6.31	29.14	28.92	32.53	30.49
Contr	7.65	7.48	34.79	34.85	40.79	40.59
Elast	14.84	14.10	43.30	43.02	40.95	40.05
Pixel	9.28	9.43	34.35	34.15	40.37	40.48
Jpeg	10.84	10.69	35.77	35.50	45.68	45.76
Average	10.74	**10.58**	38.47	**38.28**	50.98	**50.78**

the source images. However, it needs to consume some memory space to store a number of source features. So we propose another strategy that follows the setting of TTA more strictly. Specifically, we use the fitted category-wise distributions to sample and generate all synthesized samples with Eq.12 instead of generating them based on the original features (Eq.11). In this way, we only need to store the distribution parameters and not the source features themselves. Tab. VI shows the comparison results on CIFAR-10, CIFAR-100, and Imagenet-C. Using only the fitted feature distribution to generate samples can align the distributions and achieve successful performance, while using the original features can achieve better performance.

V. Conclusion

We propose a Hierarchical Distribution Alignment method for the test time adaptation (TTA) problem. We use Gaussian distributions to represent the image features and align their distributions at the category-level. We propose to mine the subcategories within each category according to the implicit semantic concepts of image features. To better adapt domains, we model and align feature distributions at the subcategory-level for fine-grained domain adaptation. We use the consistency of category and subcategory prediction labels to filter target features, thereby reducing the impact of unreliable target samples. In addition, we generate synthesized features to fully utilize the diversity of feature samples and propose to reformulate TTA as a sample-level feature matching problem. We conduct extensive experiments to verify the performance of our proposed method.

References

[1] S. Xia, S. Zheng, G. Wang, X. Gao, and B. Wang, "Granular ball sampling for noisy label classification or imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2144–2155, Apr. 2023.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[3] S. Shao, Y. Wang, B. Liu, W. Liu, Y. Wang, and B. Liu, "FADS: Fourier-augmentation based data-shunting for few-shot classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 2, pp. 839–851, Feb. 2024.

[4] G. Li, Q. Gao, J. Han, and X. Gao, "A coarse-to-fine cell division approach for hyperspectral remote sensing image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 6, pp. 4928–4941, Jun. 2024.

[5] X. Zhao, C. Li, J. Wu, and X. Li, "Riemannian manifold-based feature space and corresponding image clustering algorithms," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2680–2693, Feb. 2024.

[6] C. Liu, Z. Wu, J. Wen, Y. Xu, and C. Huang, "Localized sparse incomplete multi-view clustering," *IEEE Trans. Multimedia*, vol. 25, pp. 5539–5551, 2022.

[7] C. Liu, J. Wen, Z. Wu, X. Luo, C. Huang, and Y. Xu, "Information recovery-driven deep incomplete multiview clustering network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 1–11, Nov. 2024.

[8] J. Chen, A. Huang, W. Gao, Y. Niu, and T. Zhao, "Joint shared-and-specific information for deep multi-view clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 12, pp. 7224–7235, Dec. 2023.

[9] G. Ke, G. Chao, X. Wang, C. Xu, Y. Zhu, and Y. Yu, "A clustering-guided contrastive fusion for multi-view representation learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 4, pp. 2056–2069, Apr. 2024.

[10] C. Huang et al., "Hierarchical graph embedded pose regularity learning via spatio-temporal transformer for abnormal behavior detection," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 307–315.

[11] C. Huang, Y. Shi, J. Wen, W. Wang, Y. Xu, and X. Cao, "Ex-VAD: Explainable fine-grained video anomaly detection based on visual-language models," in *Proc. 42nd Int. Conf. Mach. Learn.*, 2025, pp. 1–12.

[12] C. Huang, W. Huang, Q. Jiang, W. Wang, J. Wen, and B. Zhang, "Multimodal evidential learning for open-world weakly-supervised video anomaly detection," *IEEE Trans. Multimedia*, vol. 27, pp. 3132–3143, 2025.

[13] G. Kang, Y. Wei, Y. Yang, Y. Zhuang, and A. Hauptmann, "Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3569–3580.

[14] Z. Zhang et al., "A componentwise approach to weakly supervised semantic segmentation using dual-feedback network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7541–7554, Oct. 2023.

[15] C. Zeng, X. Yang, M. Mirmehdi, A. Gambaruto, and T. Burghardt, "Video-TransUNet: Temporally blended vision transformer for CT VFSS instance segmentation," in *Proc. 15th Int. Conf. Mach. Vis. (ICMV)*, Jun. 2023, p. 20.

[16] W. Zhou, H. Zhang, W. Yan, and W. Lin, "MMSMCNet: Modal memory sharing and morphological complementary networks for RGB-T urban scene semantic segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 12, pp. 7096–7108, Dec. 2023.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[18] Z. Wu, C. Liu, J. Wen, Y. Xu, J. Yang, and X. Li, "Selecting high-quality proposals for weakly supervised object detection with bottom-up aggregated attention and phase-aware loss," *IEEE Trans. Image Process.*, vol. 32, pp. 682–693, 2023.

[19] Z. Wu, C. Liu, C. Huang, J. Wen, and Y. Xu, "Deep object detection with example attribute based prediction modulation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 2020–2024.

[20] Z. Wu, J. Wen, Y. Xu, J. Yang, X. Li, and D. Zhang, "Enhanced spatial feature learning for weakly supervised object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 961–972, Jan. 2024.

[21] Q. Qi, T. Hou, Y. Yan, Y. Lu, and H. Wang, "TCNet: A novel triple-cooperative network for video object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 3649–3662, Aug. 2023.

[22] Z. Xie et al., "Cross-modality double bidirectional interaction and fusion network for RGB-T salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 4149–4163, Aug. 2023.

[23] D. Wang, E. Shelhamer, S. Liu, B. A. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–15.

[24] V. VS, P. Oza, and V. M. Patel, "Towards online domain adaptive object detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 478–488.

[25] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 2. Lile, France: International Machine Learning Society (IMLS), Jul. 2015, pp. 1180–1189.

[26] S. M. Ayyoubzadeh et al., "Test-time adaptation for optical flow estimation using motion vectors," *IEEE Trans. Image Process.*, vol. 32, pp. 4977–4988, 2023.

[27] K. Wei, X. Yang, Z. Xu, and C. Deng, "Class-incremental unsupervised domain adaptation via pseudo-label distillation," *IEEE Trans. Image Process.*, vol. 33, pp. 1188–1198, 2024.

[28] X. Liu, Y. Huang, H. Wang, Z. Xiao, and S. Zhang, "Universal and scalable weakly-supervised domain adaptation," *IEEE Trans. Image Process.*, vol. 33, pp. 1313–1325, 2024.

[29] Z. Wang, L. Shen, M. Xu, M. Yu, K. Wang, and Y. Lin, "Domain adaptation for underwater image enhancement," *IEEE Trans. Image Process.*, vol. 32, pp. 1442–1457, 2023.

[30] B. Xu, Z. Zeng, C. Lian, and Z. Ding, "Few-shot domain adaptation via mixup optimal transport," *IEEE Trans. Image Process.*, vol. 31, pp. 2518–2528, 2022.

[31] J. Wang and J. Jiang, "Learning across tasks for zero-shot domain adaptation from a single source domain," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6264–6279, Oct. 2022.

[32] Y. Sun, X. Wang, Z. Liu, J. Miller, A. A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," 2019, *arXiv:1909.13231*.

[33] Y. Liu, P. Kothari, B. G. v. Delft, B. Bellot-Gurlet, T. Mordan, and A. Alahi, "TTT++: When does self-supervised test-time training fail or thrive?," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 21808–21820.

[34] H. Tang, K. Chen, and K. Jia, "Unsupervised domain adaptation via structurally regularized deep clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8722–8732, doi: 10.1109/CVPR42600.2020.00875.

[35] Y. Su, X. Xu, and K. Jia, "Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 17543–17555.

[36] X. Shen, M. Shao, S. Pan, L. T. Yang, and X. Zhou, "Domain-adaptive graph attention-supervised network for cross-network edge classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 1–14, Dec. 2024.

[37] Y. Ge, F. Zhu, D. Chen, R. Zhao, X. Wang, and H. Li, "Structured domain adaptation with online relation regularization for unsupervised person re-ID," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 258–271, Jan. 2024.

[38] C. Liu, S. Cheng, W. Ding, and R. Arcucci, "Spectral cross-domain neural network with soft-adaptive threshold spectral enhancement," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 692–703, Jan. 2025.

[39] M. Guo, B. Chen, Z. Yan, Y. Wang, and Q. Ye, "Virtual classification: Modulating domain-specific knowledge for multidomain crowd counting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 2, pp. 2958–2972, Feb. 2025.

[40] Q. Ren, Q. Mao, and S. Lu, "Prototypical bidirectional adaptation and learning for cross-domain semantic segmentation," *IEEE Trans. Multimedia*, vol. 26, pp. 501–513, 2024.

[41] T. Chen, S.-H. Wang, Q. Wang, Z. Zhang, G.-S. Xie, and Z. Tang, "Enhanced feature alignment for unsupervised domain adaptation of semantic segmentation," *IEEE Trans. Multimedia*, vol. 24, pp. 1042–1054, 2022.

[42] Y. Xu, F. He, B. Du, D. Tao, and L. Zhang, "Self-ensembling GAN for cross-domain semantic segmentation," *IEEE Trans. Multimedia*, vol. 25, pp. 7837–7850, 2023.

[43] Y. Tian, J. Li, H. Fu, L. Zhu, L. Yu, and L. Wan, "Self-mining the confident prototypes for source-free unsupervised domain adaptation in image segmentation," *IEEE Trans. Multimedia*, vol. 26, pp. 7709–7720, 2024.

[44] Y. Liu, J. Wang, L. Xiao, C. Liu, Z. Wu, and Y. Xu, "Foregroundness-aware task disentanglement and self-paced curriculum learning for domain adaptive object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 369–380, Jan. 2025.

[45] Y. Liu, J. Wang, W. Wang, Y. Hu, Y. Wang, and Y. Xu, "CRADA: Cross domain object detection with cyclic reconstruction and decoupling adaptation," *IEEE Trans. Multimedia*, vol. 26, pp. 1–12, 2024.

[46] Z. Piao, L. Tang, and B. Zhao, "Unsupervised domain-adaptive object detection via localization regression alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 1–12, Nov. 2024.

[47] X. Liu, W. Li, and Y. Yuan, "Decoupled unbiased teacher for source-free domain adaptive medical object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 7287–7298, Jun. 2024.

[48] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. NeurIPS*, Dec. 2017, pp. 1–11.

[49] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3722–3731.

[50] Y. Luo, Z. Huang, Z. Wang, Z. Zhang, and M. Baktashmotlagh, "Adversarial bipartite graph learning for video domain adaptation," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 19–27.

[51] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.

[52] C.-C. Hsu, Y.-H. Tsai, Y.-Y. Lin, and M.-H. Yang, "Every pixel matters: Center-aware feature alignment for domain adaptive object detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 733–748.

[53] C. Chen et al., "Relation matters: Foreground-aware graph-based relational reasoning for domain adaptive object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3677–3694, Mar. 2023.

[54] Y. Liu, J. Wang, C. Huang, Y. Wang, and Y. Xu, "CIGAR: Cross-modality graph reasoning for domain adaptive object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 23776–23786.

[55] K. Chen, T. Gong, and L. Zhang, "Camera-aware recurrent learning and Earth mover's test-time adaption for generalizable person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 1, pp. 357–370, Jan. 2024.

[56] M. Choi, J. Choi, S. Baik, T. H. Kim, and K. M. Lee, "Test-time adaptation for video frame interpolation via meta-learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9615–9628, Dec. 2022.

[57] Y. Su, X. Xu, T. Li, and K. Jia, "Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering regularized self-training," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5524–5540, Aug. 2024.

[58] L. Zhang, J. Nie, W. Wei, and Y. Zhang, "Unsupervised test-time adaptation learning for effective hyperspectral image super-resolution with unknown degeneration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 7, pp. 5008–5025, Jul. 2024.

[59] R. A. Marsden, M. Döbler, and B. Yang, "Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 2543–2553.

[60] R. Wen, H. Yuan, D. Ni, W. Xiao, and Y. Wu, "From denoising training to test-time adaptation: Enhancing domain generalization for medical image segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Jan. 2024, pp. 464–474.

[61] J. Liang, R. He, and T. Tan, "A comprehensive survey on test-time adaptation under distribution shifts," *Int. J. Comput. Vis.*, vol. 133, no. 1, pp. 31–64, Jan. 2025.

[62] S. Zhang et al., "Tracking persons-of-interest via unsupervised representation adaptation," 2017, *arXiv:1710.02139*.

[63] Y. Yu, L. Sheng, R. He, and J. Liang, "STAMP: Outlier-aware test-time adaptation with stable memory replay," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 375–392.

[64] J. N. Kundu, S. Bhambri, A. Kulkarni, H. Sarkar, V. Jampani, and R. V. Babu, "Concurrent subsidiary supervision for unsupervised source-free domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 177–194.

[65] A. Karmanov, D. Guan, S. Lu, A. El Saddik, and E. Xing, "Efficient test-time adaptation of vision-language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 14162–14171.

[66] M. Shu et al., "Test-time prompt tuning for zero-shot generalization in vision-language models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 14274–14289.

[67] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation," in *Proc. Comput. Vis. Pattern Recognit. Comput. Vis. Pattern Recognit.*, Feb. 2020, pp. 6028–6039.

[68] Y. Liu, J. Wang, C. Huang, Y. Wu, Y. Xu, and X. Cao, "MLFA: Toward realistic test time adaptive object detection by multi-level feature alignment," *IEEE Trans. Image Process.*, vol. 33, pp. 5837–5848, 2024.

[69] Y. Iwasawa and Y. Matsuo, "Test-time classifier adjustment module for model-agnostic domain generalization," in *Proc. Neural Inf. Process. Syst.*, vol. 34, Dec. 2021, pp. 2427–2440.

[70] C. Liu, J. Wen, X. Luo, and Y. Xu, "Incomplete multi-view multi-label learning via label-guided masked view- and category-aware transformers," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2023, pp. 8816–8824.

[71] W. Li, X. Liu, and Y. Yuan, "SIGMA: Semantic-complete graph matching for domain adaptive object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5281–5290.

[72] D. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[73] K. Fu, S. Liu, X. Luo, and M. Wang, "Robust point cloud registration framework based on deep graph matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8889–8898, doi: 10.1109/CVPR46437.2021.00878.

[74] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," 2019, *arXiv:1903.12261*.

[75] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[76] M. J. Mirza, J. Micorek, H. Possegger, and H. Bischof, "The norm must go on: Dynamic unsupervised domain adaptation by normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14745–14755, doi: 10.1109/CVPR52688.2022.01435.

[77] S. Niu et al., "Efficient test-time model adaptation without forgetting," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 16888–16905.

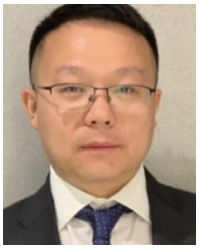
[78] M. Boudiaf, R. Mueller, I. B. Ayed, and L. Bertinetto, "Parameter-free online test-time adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8334–8343.

[79] Z. Zhou, L.-Z. Guo, L.-H. Jia, D. Zhang, and Y.-F. Li, "ODS: Test-time adaptation in the presence of open-world data shift," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 42574–42588.
[80] J. Shin and H. Kim, "L-TTA: Lightweight test-time adaptation using a versatile stem layer," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 37, 2024, pp. 39325–39349.
[81] D. Osowiechi et al., "Watt: Weight average test time adaptation of clip," in *Proc. 38th Annu. Conf. Neural Inf. Process. Syst.*, 2024, pp. 48015–48044.
[82] J. Ma, "Improved self-training for test-time adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 23701–23710.
[83] S. Niu et al., "Towards stable test-time adaptation in dynamic wild world," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023, pp. 1–14.
[84] M. Jang and H. W. Chung, "Label distribution shift-aware prediction refinement for test-time adaptation," in *Proc. Trans. Mach. Learn. Res.*, Feb. 2025.
[85] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019, doi: 10.1145/3326362.
[86] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2013, pp. 245–248.

Yabo Liu received the Ph.D. degree in computer technology from Harbin Institute of Technology, Shenzhen, China, in 2025. From 2021 to 2025, he was a jointly supervised Ph.D. candidate by Harbin Institute of Technology and Peng Cheng Laboratory. He is currently a Lecturer with the School of Artificial Intelligence, Ocean University of China, Qingdao, China. His dissertation was nominated for Harbin Institute of Technology's Outstanding Dissertation Award. His current research interests include computer vision, transfer learning, machine learning, and multi-modal learning.

Chao Huang (Member, IEEE) received the Ph.D. degree in computer science and technology from Harbin Institute of Technology, Shenzhen, China, in 2022. From 2019 to 2022, he was a Visiting Student with the Peng Cheng Laboratory. He is currently an Assistant Professor with the School of Cyber Science and Technology, Sun Yat-sen University. So far, he has published over 20 technical papers at prestigious international journals and conferences. His research interests include anomaly detection, multimedia analysis, object detection, image/video compression, and deep learning. He received the Distinguished Paper Award of AAAI 2023, and his dissertation was nominated for Harbin Institute of Technology's Outstanding Dissertation Award. He serves as an Associated Editor for the *International Journal of Image and Graphics*, and also serves/served as the PC member for several top conferences, including CVPR, ICCV, NeurIPs, ICLR, AAAI, ACM Multimedia, and IEEE ICME.

Yong Xu (Senior Member, IEEE) received the Ph.D. degree in pattern recognition and intelligence system form the Nanjing University of Science and Technology, Nanjing, China, in 2005. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He has authored or co-authored more than 70 papers in top tier academic journals and conferences. His articles have been cited more than 5,800 times in the Web of Science, and 15,000 times in the Google Scholar. His research interests include pattern recognition, deep learning, biometrics, machine learning, and video analysis. He was the Co-Editor-in-Chief of the *International Journal of Image and Graphics*, an Associate Editor for *CAAI Transactions on Intelligence Technology*, and the Editor of the *Pattern Recognition* and *Artificial Intelligence*.



Xiaochun Cao (Senior Member, IEEE) received the B.E. and M.E. degrees in computer science from Beihang University (BUAA), China, and the Ph.D. degree in computer science from the University of Central Florida, USA, with his dissertation nominated for the university level Outstanding Dissertation Award. After graduation, he spent about three years as a Research Scientist with ObjectVideo Inc. From 2008 to 2012, he was a Professor with Tianjin University. Before joining SYSU, he was a Professor with the Institute of Information Engineering, Chinese Academy of Sciences. He is currently a Professor and the Dean of the School of Cyber Science and Technology, Sun Yat-sen University. He has authored and co-authored over 200 journals and conference papers. In 2004 and 2010, he was the recipient of the Piero Zamperoni Best Student Paper Award at the International Conference on Pattern Recognition. He is on the editorial boards of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and IEEE TRANSACTIONS ON IMAGE PROCESSING, and was on the editorial boards of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and IEEE TRANSACTIONS ON MULTIMEDIA.

Jinghua Wang (Member, IEEE) received the B.S. degree from Shandong University in 2005, the M.S. degree from Harbin Institute of Technology in 2009, and the Ph.D. degree from The Hong Kong Polytechnic University in 2013. From 2014 to 2016, he was a Research Fellow with Nanyang Technological University, Singapore. From 2017 to 2022, he was a Research Assistant Professor with Shenzhen University. He is currently an Associate Professor with the School of Computer Science and Software Engineering, Harbin Institute of Technology (Shenzhen), China. His current research interests include computer vision and machine learning.