Un-mixing Test-time Adaptation under Heterogeneous Data Streams

Zixian Su*, Jingwei Guo*, Xi Yang†, Qiufeng Wang, Kaizhu Huang†

***Abstract*—Deploying deep models in real-world scenarios remains challenging due to significant performance drops under distribution shifts between training and deployment environments. Test-Time Adaptation (TTA) has recently emerged as a promising solution, enabling on-the-fly model adaptation without access to source data. However, its effectiveness degrades significantly in the presence of complex, mixed distribution shifts – common in practical settings – where multiple latent domains coexist. Adapting under such intrinsic heterogeneity, especially in unlabeled and online conditions, remains an open and underexplored challenge. In this paper, we study TTA under mixed distribution shifts and move beyond conventional homogeneous adaptation paradigms. By revisiting TTA from a frequency-domain perspective, we observe that distribution heterogeneity often manifests in Fourier space – for instance, high-frequency components tend to carry domain-specific variations. This motivates us to perform domain-aware separation using high-frequency texture cues, making diverse shift patterns more tractable. To this end, we propose FreDA, a novel Frequency-based Decentralized Adaptation framework that decomposes globally heterogeneous data into locally homogeneous components in the frequency domain. It further employs decentralized learning and augmentation strategies to robustly adapt under complex, evolving shifts. Extensive experiments across various environments (corrupted, natural, and medical) demonstrate the superiority of our proposed framework over the state-of-the-arts.**

***Index Terms*—Test-time Adaptation, Transfer Learning**

I. Introduction

DEEP learning models often suffer significant performance degradation when deployed in environments where the data distribution differs from that of the training set – a challenge known as domain shift [1], [2]. Recently, Test-Time Adaptation (TTA) [3]–[10] has emerged as a promising solution by refining model parameters to better align with the encountered data at inference time. It leverages the incoming data stream for real-time adjustments without the need for retraining on a labeled dataset, enabling swift model adaptation to unpredictable data characteristics during deployment.

Despite their success, current TTA models often are limited to ideal testing conditions, typically involving homogeneous test samples with similar types of distribution shifts. In reality, distribution shifts are *mixed, overlapping, and even conflicting* [11]–[15]. For instance, photo management software handles diverse corruptions, such as noise, blur, compression; medical imaging platforms manage inconsistencies from varied acquisition methods; and autonomous driving systems face fluctuating conditions like lighting, weather, and road types.

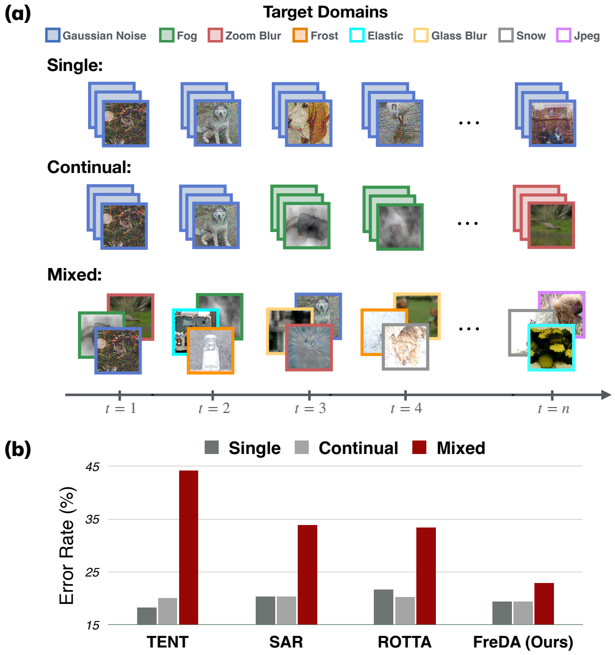


Fig. 1: (a) Illustration of three TTA scenarios with single, continual, and mixed domain shifts. (b) Classification error rate on CIFAR-10-C across different TTA settings: performance of conventional models drops sharply under mixed domain shifts.

Such complexity poses serious challenges for existing methods. While recent efforts have extended TTA to continually changing environments [5], they typically assume a uniform target domain at each time step, as illustrated in the continual setting in Figure 1 (a). Some methods periodically reset the model to its source pre-trained state [7], [9] to mitigate the cumulative effect of sequential distribution shifts on adaptation. Others down-weight outlier target samples [6], [10] to suppress the impact of abrupt or rare shifts. Although these strategies aim to improve robustness under dynamic conditions, they fall short when facing the entangled / mixed real-world distributions (see Figure 1 (b)). Their shortcomings stem from a shared limitation: they lack the capacity to disentangle or localize domain variations. As a result, they often overfit to dominant patterns or suffer from catastrophic forgetting, failing to be deployed in realistic, heterogeneous data streams.

Capturing such intrinsic heterogeneity under unlabeled, online conditions is particularly challenging, and effective adaptation in this setting remains an open problem. To tackle

Zixian Su is with Beijing Academy of Artificial Intelligence, Beijing, China and University of Liverpool, Liverpool, UK (E-mail: zxsu@baai.ac.cn); Jingwei Guo is with Alibaba Group, Beijing, China and University of Liverpool, Liverpool, UK (E-mail: jingweiguo19@outlook.com); Xi Yang and Qiufeng Wang are with Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, China (Email: xi.yang01@xjtlu.edu.cn; qiufeng.wang@xjtlu.edu.cn); Kaizhu. Huang is with Duke Kunshan University, Kunshan, Jiangsu, China (Email: kaizhu.huang@dukekunshan.edu.cn).

*Equal contribution.

†Corresponding authors.

this, we propose shifting from coarse, homogeneous adaptation to a fine-grained strategy that explicitly disentangles heterogeneous shifts. Our approach leverages the frequency information to disentangle domain variations, which naturally separates data features across frequency bands: high-frequency components capture fine details like edges and textures, while low-frequency components represent global structures such as shape and illumination (Section IV). By decomposing inputs accordingly, we better characterize distributional variations and identify diverse shifts. Moreover, the Fourier transform operates directly on raw pixel-level inputs, which shields the method from performance degradation under large domain gaps, and also enhancing adaptability and deployment efficiency in real-world applications.

Building upon this insight, we introduce a framework, termed Frequency-based Decentralized Adaptation (FreDA). FreDA begins by partitioning incoming data in the Fourier domain, where high-frequency components are used to group samples with similar shift characteristics. This transforms globally heterogeneous inputs into locally homogeneous subsets before any adaptation takes place. Based on these partitions, FreDA deploys multiple local models, each assigned to a specific subset. These models adapt independently to their respective data streams while periodically synchronizing through parameter exchange. The aggregated parameters form a shared base model, which is then used to reinitialize the local models, ensuring continual and coordinated adaptation across domains. This approach not only alleviates clustering errors via collaborative knowledge sharing but also allows local models to capture diverse distribution shifts. To further enhance robustness, we introduce a novel Fourier-based augmentation scheme that improves sample quality and strengthens adaptation to shift-specific characteristics.

To summarize, the main contributions of this work are three-fold:

- We identify a key limitation of most existing TTA methods – their neglect of real-world data heterogeneity – which leads to suboptimal performance when confronted with mixed and diverse distribution shifts.
- We propose FreDA, a frequency-based decentralized adaptation framework that leverages spectral decomposition and localized adaptation to effectively address heterogeneous distribution shifts at test-time.
- We validate FreDA through extensive experiments on corrupted, natural, and medical benchmarks, demonstrating consistent improvements over state-of-the-art methods across various TTA scenarios.

II. Connections to Previous Studies

A. Transfer Learning, Domain Adaptation, Test-time Training, and Test-time Adaptation

Deep neural networks often experience performance degradation when deployed in environments different from their training settings, due to distribution shifts between source (training) and target (testing) domains. This issue is extensively studied under the umbrella of transfer learning [16], [17], which aims to transfer knowledge across domains. A key branch of transfer learning is domain adaptation (DA) [18]–[21], where models trained on a labeled source domain are adapted to a target domain. Depending on the availability of source and target data, DA can be classified into: Supervised DA using labeled data in both source and target domains during training; Unsupervised DA, which relies on labeled source data and unlabeled target data; and Source-free DA, that adapts to the target domain without access to source data, typically due to privacy or transmission constraints.

TABLE I: Comparison of different transfer learning settings.

Topic	Source Data	Target Labels	Online Adaptation	Model Agnostic
Supervised Domain Adaptation	✓	✓	×	–
Unsupervised Domain Adaptation	✓	×	×	–
Source-free Domain Adaptation	×	×	×	–
Test-time Training	×	×	✓	×
Test-time Adaptation	×	×	✓	✓

Note: TTA is the most practical setting for real-world deep model deployment.

While traditional DA methods assume access to the whole target distribution during adaptation, such access is not always possible in real-world deployment. To address this, two prominent test-time adaptation paradigms have been proposed: Test-time Training (TTT) [22]–[24] introduces an auxiliary task – often self-supervised – during pre-training. At test time, the model optimizes this auxiliary objective to adapt to the target distribution accordingly; Test-time Adaptation (TTA) [3], [7], [8] poses a more demanding and practical scenario, where the model must adapt on-the-fly to the test stream without any prior modification during training. This setting emphasizes the need for rapid, real-time model updates to effectively capture the continuously incoming data. A brief comparison of these adaptation settings is summarized in TABLE I. Our work falls under the TTA setting, where we propose a generalizable and robust adaptation framework that maintains strong performance across diverse distribution shifts and corruptions – without relying on source data or altering the pre-training process.

B. Non-i.i.d. Test-time Adaptation

While conventional TTA methods primarily focus on single domain shifts under the idealized independent and identically distributed (i.i.d.) assumption, real-world deployment often violates these conditions due to inherent data heterogeneity. This discrepancy has motivated recent efforts to extend TTA to more realistic settings, with two primary challenges:

1) Mixed Domains: Modern TTA methods designed for continual domain shifts [5], [6], [9], [10], [25] have pushed beyond the single adaptation setting. However, these approaches assume that the target domain is uniform at each time point – a special case of mixed domains where the change occurs across distinct time periods. In reality, data streams often involve genuinely mixed domains with heterogeneous distributions present simultaneously. Existing approaches primarily focus on stabilizing model updates through periodic parameter

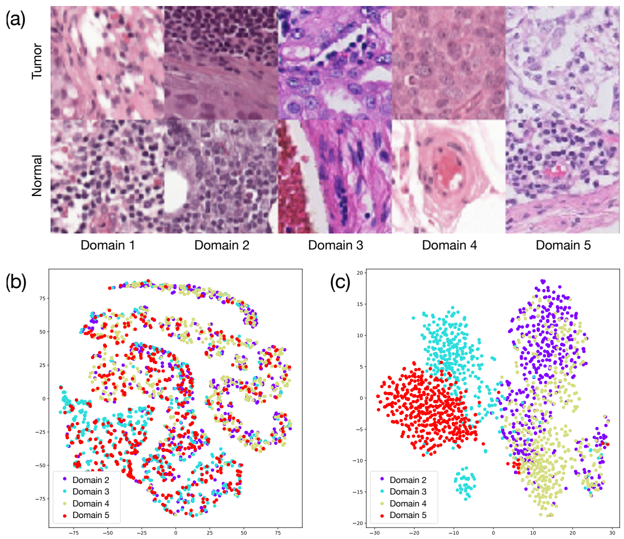
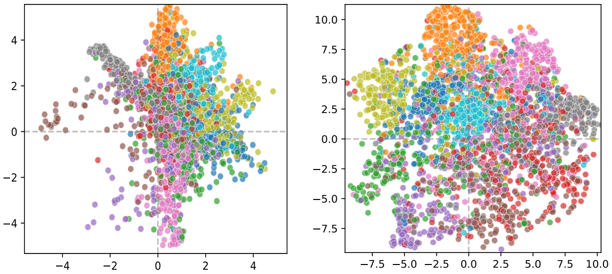


Fig. 2: (a) Visual heterogeneity across five healthcare centers in Camelyon17 dataset: Example patches showcase diverse domain shifts that characterize mixed target domains. (b) Conventional sample latent features from pretrained models fail to separate different target subdomains, showing significant overlap. (c) High-frequency information enable distinct separation of target subdomains, contrasting with the conventional ones.

resets [7], [9] or importance weighting of target samples [6], [9], [10], but these techniques are ill-equipped to disentangle and manage the complex interleaving of domain shifts, leading to degraded performance. In contrast, our work adopts a data-centric perspective by explicitly addressing mixed domains via frequency-space decomposition, which facilitates domain separation and proactive distribution alignment prior to model adaptation. While recent efforts like [7] acknowledge mixed distribution challenges under the broader "Dynamic Wild World" paradigm, their unified treatment of diverse real-world factors does not systematically address data heterogeneity. Conversely, our study directly targets the core issue of mixed domains in TTA, introducing tailored strategies for disentangling and adapting to co-occurring heterogeneous distributions.

2) Dependent Sampling: The second challenge stems from class-level dependencies introduced by temporal data sampling. This topic has garnered significant research attention, with a growing body of work [25]–[30] actively addressing it through strategies such as pseudo-label-based rebalancing, or extended observation windows. These approaches aim to correct the temporal imbalance caused by skewed class distributions over time. However, unlike this these studies that focus on class-level imbalance, our work diverges by targeting sample-level heterogeneity, which remains underexplored. Specifically, we address style diversity and distributional variation at test-time – a challenge that persist even under class-balanced conditions – to improve model robustness under mixed-domain scenarios.

Methods	C10	C100	IN
TBN (Centralized)	33.8	45.8	82.5
TBN (Decentralized)	28.5	43.2	77.6



(a) Centralized Adaptation (b) Decentralized Adaptation

Fig. 3: Experimental Result (error rate) and t-SNE feature visualization using TBN [31] as the pseudo-labeling method under mixed distribution shifts, comparing centralized and decentralized adaptation. Different colors represent different classes. (a) Centralized Adaptation: Global BN parameters are applied to the entire batch. (b) Decentralized Adaptation: Localized BN parameters are tailored for clusters separated based on high-frequency features. C10, C100, IN respectively denotes CIFAR-10-C, CIFAR-100-C and ImageNet-C datasets.

III. Problem Definition of mixed domain shifts

Test-time adaptation (TTA) aims to adjust a model $q_\theta(y|x)$, initially trained on a source dataset $\mathcal{D}_s = \{(x, y) \sim p_s(x, y)\}$, to a target domain $\mathcal{D}_t = \{(x, y) \sim p_t(x, y)\}$ on a data stream without accessing source data or target labels. TTA handles covariate shift by assuming $p_s(y|x) = p_t(y|x)$ while $p_s(x) \neq p_t(x)$. This challenge intensifies when $\mathcal{D}_t$ contains multiple non-i.i.d sub-distributions $p_{t_i}(x)$, such as

$$p_t(x) \leftarrow \{p_{t_1}(x), p_{t_2}(x), \ldots, p_{t_N}(x)\}.$$

Specifically, we have $\mathcal{D}_t = \mathcal{D}_{t_1} \cup \mathcal{D}_{t_2} \cup \cdots \cup \mathcal{D}_{t_N}$ where $x \in \mathcal{D}_{t_1}$ satisfying $x \sim p_{t_1}(x)$. This scenario requires the model $q_\theta(y|x)$ to effectively handle the heterogeneous and evolving target distribution to maintain robust performance. TTA strategies must therefore refine the model to optimize its predictive accuracy across these diverse sub-domains, ensuring consistent and reliable performance amidst significant distributional variability.

IV. TTA under Mixed Distribution Shifts: A Fourier Perspective

While Test-Time Adaptation (TTA) methods excel under single-type distribution shifts, their performance degrades catastrophically when facing mixed distribution shifts – a ubiquitous challenge in real-world deployment. As demonstrated on CIFAR-10-C (Figure 1 (b)), leading methods like TENT [3], SAR [7] and RoTTA [25] suffer 16% average accuracy drop compared to single-shift scenarios. Conventional TTA paradigms attempt to mitigate this through passive robustness measures, such as entropy regularization [3], gradient masking [7], or model resetting [9]. These methods treat mixed

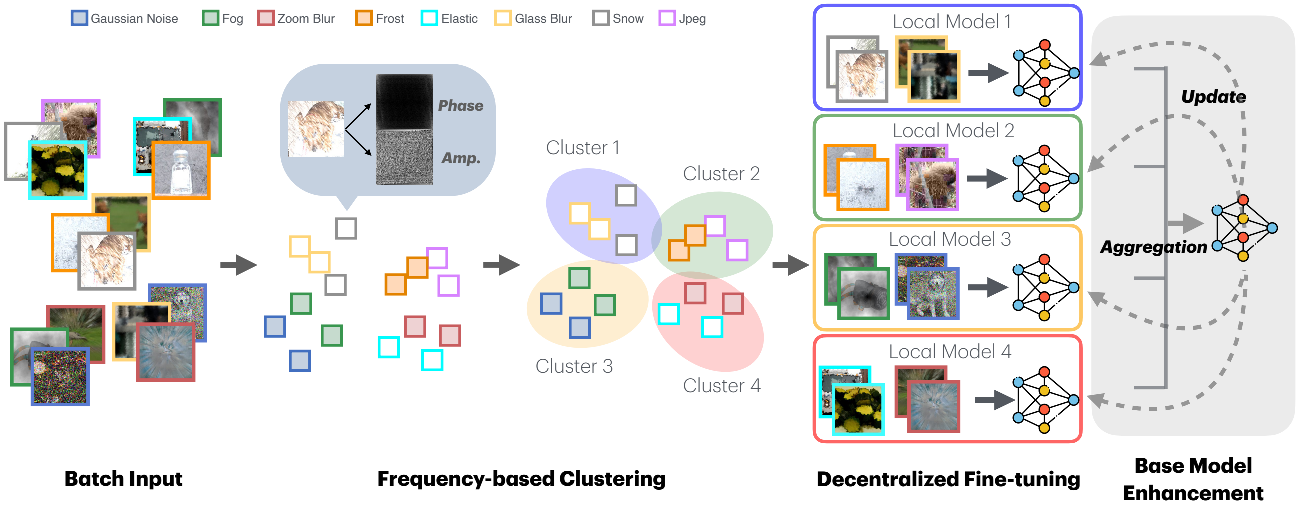


Fig. 4: Illustration of Frequency-based Decentralized Learning.

distribution shifts as irreducible noise, deferring corrective actions until model failures occur – a reactive paradigm that inevitably compromises adaptation efficacy.

To address this, we propose a paradigm shift to *proactive distribution management* – instead of demanding models to handle arbitrary mixtures, we first resolve the heterogeneity via frequency-space decomposition. Our key insight is that mixed distributions exhibit spectral signatures that can be disentangled before model adaptation begins. As evidenced in Figure 2 (c), high-frequency components naturally separate distinct shifts without relying on error-prone model features (see Figure 2 (b). This theoretically grounded decomposition converts the ill-posed mixed-shift adaptation into multiple well-conditioned homogeneous-shift problems, effectively create an ideal adaptation environment from chaotic real-world streams. Crucially, this decomposition enables a *decentralized adaptation* where we can deploy domain-specific models tailored to high-frequency-separated clusters. Take TBN [31] as an example: while its centralized variant recalculates batch normalization (BN) statistics globally (treating heterogeneous data as a single distribution), our decentralized adaptation computes localized BN parameters within each spectrally disentangled cluster. As shown in Figu 3, this context-aware estimation preserves semantic discriminability: samples exhibit tighter within-class cohesion and sharper between-class boundaries compared to the blurred separability of *centralized adaptation*. Building on these insights, the following section details how leveraging the frequency domain enhances TTA methods for realistic scenarios with mixed distribution shifts.

V. Frequency-based Decentralized Adaptation

The previous discussion highlights how heterogeneity within target distributions can hinder effective model adaptation. This naturally raises the question: *How can we manage such distributional heterogeneity to enable better adaptation?* As outlined earlier, distinguishing samples associated with different distribution shifts offers a promising pathway. Building on this insight, we address the TTA problem by leveraging high-frequency components in the data and propose a novel framework termed Frequency-based Decentralized Adaptation (FreDA). It partitions target samples into multiple homogeneous subdomains in the Fourier space, facilitating more accurate model adaptation. This is further enhanced by a frequency-based augmentation strategy that enriches each subdomain with diverse samples, strengthening model robustness.

A. Frequency-based Decentralized Learning

Fourier transform offers an effective method to extract different frequency components from images, with high-frequency information particularly useful for capturing fine-grained details such as texture and noise. These details often highlight subtle variations among different distribution shifts. Based on this insight, we propose a module termed Frequency-based Decentralized Learning. It leverages frequency information extracted from the pixel space to systematically partition data into multiple homogeneous subsets, enabling multiple local models to specialize in capturing each distribution shift individually. Concurrently, our method enables collaborative learning by allowing periodic parameter sharing among these local models via an enhanced base model, thereby boosting overall adaptability to diverse distribution shifts.

1) Frequency Feature Extraction: We start by extracting frequency domain features from the input images. Let $\mathbf{X} \in \mathbb{R}^{n \times c \times h \times w}$ denote a batch of input images, where n is the batch size, c is the number of channels, h and w are the height and width. We first apply a Fourier transform $\mathcal{F}$ to each image $\mathbf{x}_i$. This transform converts the image from the spatial domain to the frequency domain, producing a complex-valued representation $\mathcal{F}(\mathbf{x}_i) \in \mathbb{C}^{h \times w \times c}$ that contains both real $R(\mathbf{x}_i)$ and imaginary part $I(\mathbf{x}_i)$. Next, we compute the amplitude spectrum $A(\mathbf{x}_i)(u,v)$ using $A(x)(u,v) = \sqrt{R^2(x)(u,v) + I^2(x)(u,v)}$. It reveals the intensity of the frequency content, e.g., high-frequency amplitudes highlight edges and fine details while low-frequency amplitudes emphasize the overall structure and gradual changes. Then, we filter out low-frequency elements using mask $M(u,v) = \mathbb{1}\left(\left(u < \frac{h}{4} \vee u > \frac{3h}{4}\right) \vee \left(v < \frac{w}{4} \vee v > \frac{3w}{4}\right)\right)$ to emphasize the high-frequency components $G(x)(u,v)$ that are more likely to indicate shifts in distribution:

$$G(x)(u,v) = A(x)(u,v) \cdot M(u,v). \tag{1}$$

2) Frequency-Based Clustering: We then employ a clustering algorithm (e.g., K-means) to partition the frequency features into K clusters, each corresponding to a different type of distribution shift. The process is formalized as:

$$\min_{\mathbf{C},\mathbf{Z}} \sum_{i=1}^{n} \left\| \mathbf{A}_{hf,i} - \mathbf{C}_{\mathbf{Z}_i} \right\|_2^2, \tag{2}$$

where $\mathbf{A}_{hf,i} = \text{vec}(G(x_i))$, $\mathbf{C} \in \mathbb{C}^{K \times d}$, $\mathbf{Z} \in \{1, \ldots, K\}^n$ denotes the 1D high-frequency component of the amplitude spectrum, the centroids of the clusters and the cluster assignments for each image. hf refers to high-frequency components, and $d = h \times w \times c$ is flattened dimension.

3) Decentralized Fine-tuning: Test-time fine-tuning is then decentralized across these clusters, allowing for specialized adaptation within each subgroup: For each cluster k, we adapt a specialized model $q_{\theta_k}(y|x)$ that is fine-tuned using only the data within that cluster:

$$\theta_k^* = \arg\min_{\theta_k} \mathbb{E}_{x \sim p_{t,k}} \left[\mathcal{L}(q_{\theta_k}(x)) \right], \tag{3}$$

where $p_{t,k}$ represents the data distribution within cluster k, and $\mathcal{L}$ is the loss function. The predictions of each iteration are collected and sorted after the local fine-tuning.

4) Base Model Enhancement: To integrate knowledge from all subnetworks and prevent degradation on specific subdomains, we periodically aggregation their parameters at intervals of time T:

$$\theta_{\text{base}} = \sum_{k=1}^{K} \left(\frac{|\mathcal{D}_k|}{\sum_{j=1}^{K} |\mathcal{D}_j|} \theta_k \right), \tag{4}$$

where $|\mathcal{D}_k|$ denotes sample number in cluster k. This aggregation step combines the parameter updates from each subnetwork proportionally to its cluster size. The updated parameters θ_{base} are then distributed back to each subnetwork, initializing them for the next batch of training: $\theta_k \leftarrow \theta_{\text{base}}$.

B. Frequency-based Augmentation

While decentralized learning effectively mitigates batch-level heterogeneity, it may falls short in accurately characterizing individual distribution shifts – largely due to limited batch size and noise introduced by coarse clustering. To improve target data quality, TTA methods commonly adopt data augmentation as a practical strategy to enhance model generalization. However, conventional augmentation techniques – typically borrowed from standard computer vision tasks such as rotation, cropping, and mixup – are primarily tailored for single-shift scenarios. They often fail to provide the targeted, distribution shifts-aware augmentation as required under mixed domain shifts. To this end, we propose a frequency-based augmentation strategy tailored for TTA in mixed-shift environments. Specifically, our method perturbs the amplitude components of target samples in the Fourier space, enabling targeted augmentation with respect to each sample's underlying distribution. By manipulating the frequency-domain characteristics, our approach generates diverse yet distribution-consistent variations, thereby enriching subdomain representations and facilitating robust model adaptation

Algorithm 1 Framework of Frequency-based Decentralized Learning and Augmentation

```
Require: Step $t$, Input batch $\mathbf{X} = \{x_1, x_2, ..., x_n\} \in \mathbb{R}^{n \times h \times w \times c}$,
    Pretrained source model $q_\theta$, Initialize Feature Repository and Local
    Sample Pool $\mathcal{R}, \mathcal{S}_k \leftarrow \emptyset$, CLUSTER_NUM $K$, KMEANS_SIZE $N$,
    COMM_INTERVAL $f$;
    Step 1: Extract Frequency Features
for $i = 1$ to $n$ do
    $\mathbf{A}_{hf,i} \leftarrow \text{vec}(\text{G}(\mathbf{x}_i))$          ▷ Extract high-freq components
end for
    Step 2: Dynamic Clustering
$\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathbf{A}_{hf,i}\}_{i=1}^{n}$          ▷ Frequency Information Repository
$\mathcal{R} \leftarrow \mathcal{R}[(|\mathcal{R}| - N + 1) :]$          ▷ Keep the last $N$ entries for kmeans
    clustering
$(\mathbf{C}_t, \mathbf{Z}) \leftarrow$ K-means$(\mathcal{R}, K, \mathbf{C}_{t-1})$          ▷ Obtain Cluster Labels
    $\mathbf{Z} = \{Z_i\}_{i=1}^{n}$ (Eq. 2)
    Step 3: Local Model Training
for cluster $k \in \{1, \ldots, K\}$ do
    $\mathcal{S}_k \leftarrow \mathcal{S}_k \cup \{x_i \mid Z_i = k\}$          ▷ Gather samples for cluster k
    $\mathcal{S}_k \leftarrow \mathcal{S}_k[(|\mathcal{S}_k| - n + 1) :]$ ▷ Keep the last batch_size $= n$ entries
    $\mathcal{S}'_k \leftarrow$ select_samples$(\mathcal{S}_k)$          ▷ Select samples (Eq. 5)
    for each $x_i \in \mathcal{S}'_k$ do
        $\tilde{x}_i \leftarrow$ augment$(x_i)$          ▷ Augment data (Eq. 7)
        Train$(q_{\theta_k}, x_i, \tilde{x}_i)$          ▷ Train local model (Eq. 3)
    end for
end for
    Step 4: Compile Predictions
$\mathbf{Y} \leftarrow$ collect_sort$(\{q_{\theta_k}(\mathbf{X})\})$          ▷ Collect and sort predictions
    Step 5: Base Model Enhancement
If t % $f$ == 0 :          ▷ Model Communication with interval $f$ (Eq.4)
    $\theta_{\text{global}} \leftarrow \sum_{k=1}^{K} w_k \theta_k$
    $\theta_k \leftarrow \theta_{\text{global}}$
```

1) Sample Selection Mechanism: We first select the reliable samples in each local model leveraging a criterion derived from the weighted entropy framework used in ETA [6] based on two primary conditions:

$$\text{Cri} = \mathbb{1}\left[(H(\mathbf{y}_t) < H_0) \wedge (|\cos(\mathbf{y}_t, \bar{\mathbf{y}}_{t-1})| < \epsilon) \right]. \tag{5}$$

The entropy $H(\mathbf{y}_t)$ measures the uncertainty in the current predictions. The cosine similarity $\cos(\mathbf{y}_t, \bar{\mathbf{y}}_{t-1})$ denotes the deviation between the current sample's class probabilities $\mathbf{y}_t$ and the aggregated class probabilities $\bar{\mathbf{y}}_{t-1}$. ϵ is the threshold for cosine similarity, and H_0 is the fixed entropy threshold. This ensures that selected samples exhibit significant deviations from previous predictions in class distribution and lower prediction uncertainty.

2) Frequency-Based Augmentation: The augmentation process involves perturbing the amplitude spectrum. Let $A(x_i)$ represent the amplitude spectrum of a selected sample x_i. To generate a perturbed amplitude spectrum $\tilde{A}(x_i)$, we apply a random Gaussian perturbation:

$$\tilde{A}(x_i) = (1 + \alpha \cdot \Delta) \cdot A(x_i), \tag{6}$$

where $\Delta \sim \mathcal{N}(0, \sigma^2)$ is a perturbation matrix sampled from a Gaussian distribution, and α is a scaling factor. Then, the augmented sample $\tilde{x}_i$ is reconstructed via the inverse Fourier transform $\mathcal{F}^{-1}$ to the perturbed amplitude spectrum, combined with the original phase spectrum $P(x_i)$:

$$\tilde{\mathbf{X}}_i = \mathcal{F}^{-1}\left(\tilde{A}(x_i), P(x_i)\right). \tag{7}$$

3) Loss Function: The training objective combines the entropy loss of the selected samples with a consistency loss from the augmented samples. The total loss is defined as:

$$\mathcal{L}_{\text{total}} = \frac{1}{n}\sum_{i=1}^{n} H(\mathbf{y}_i) + \lambda \cdot \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_{\text{con}}\left(\hat{\mathbf{y}}_i, \tilde{\mathbf{y}}_i\right), \tag{8}$$

where the entropy loss $H(\mathbf{y}_i)$ for the original sample x_i is given by $H(\mathbf{y}_i) = -\sum_{j=1}^{C} \mathbf{y}_{i,j} \log \mathbf{y}_{i,j}$ with $\mathbf{y}_i$ being the predicted probability over the C classes, and the consistency loss $\mathcal{L}_{\text{con}}\left(\hat{\mathbf{y}}_i, \tilde{\mathbf{y}}_i\right) = -\sum_{j=1}^{C} \hat{\mathbf{y}}_{i,j} \log \tilde{\mathbf{y}}_{i,j}$ is defined as the cross-entropy between the prediction $\tilde{\mathbf{y}}_i$ of the augmented sample $\tilde{x}_i$ and the pseudo-label $\hat{\mathbf{y}}_i$ from the original sample.

C. Overall Framework

We provide the overall pipeline algorithm of FreDA in Algorithm 1. During implementation, we leverage a memory bank strategy [4], [25], [32], [33] that is updated in real time. This design serves two purposes: **1)** to ensure accurate clustering—since an overly small batch could impede the effective separation of data – and **2)** to maintain the number of samples processed by the local model consistent with the original batch size, thereby preventing performance degradation due to a drastic reduction in batch size (e.g., reducing by a factor of the cluster number).

VI. Theoretical Insights

To facilitate theoretical understanding, we leverage the expansion-based analysis framework proposed in [34]. We first revisit the core concepts and then highlight the theoretical advantages of our method under mixed domain shifts.

A. Expansion Theory

Definition 1 ((a, c)-expansion)**.** *A class-conditional distribution P_i satisfies (a, c)-expansion if $\forall S \subseteq \mathcal{X}$ with $P_i(S) \leq a$:*

$$\mathcal{P}_i(\mathcal{N}(S)) \geq \min(c\mathcal{P}_i(S), 1)$$

where $\mathcal{N}(S)$ refers to the neighborhood of S under data augmentations.

Definition 2 (Separation)**.** *P is (μ, r)-separated if:*

$$\mathbb{E}_{x\sim\mathcal{P}}\left[\max_{x'\in B_r(x)} \mathbf{1}(G(x) \neq G(x'))\right] \leq \mu$$

where $B_r(x)$ is an ℓ_2-ball of radius r.

The core theorem from [34] states:

Theorem 1 (Pseudo-label Denoising)**.** *Under (a, c)-expansion and (μ, r)-separation, any classifier G of:*

$$\min_{G} \frac{c+1}{c-1}\mathcal{L}_{pl}(G) + \frac{2c}{c-1}\mathcal{R}_B(G)$$

achieves error:

$$\text{Err}(G) \leq \frac{2}{c-1}\text{Err}(G_{pl}) + \frac{2c}{c-1}\mu$$

where $\mathcal{L}_{pl}$ is pseudo-label loss and $\mathcal{R}_B$ is consistency regularizer.

B. Analysis of Our Method

Our method reduces both terms in Theorem 1 to achieve a tighter bound through two mechanisms:

1) Frequency-Coherent Partitioning.: Let $\{P_{t_k}\}_{k=1}^{K}$ be the K sub-domains identified by frequency-based partitioning. For measurable sets $\{S_k\}_{k=1}^{K}$ where $S_k \subseteq X_k$ (the support of P_{t_k}), define:

$$\text{Err}(G_{pl}^{k}) \leq \text{Err}(G_{pl}) - \Delta_k$$

where $\Delta_k = \mathcal{P}_{t_k}(E_c)$ and $E_c = \{x \in X_k \mid G_{pl}(x) \neq y(x)\}$. This contributes to the reduction of the first term in Theorem 1, which is further supported by experimental results in Figure 3.

2) Augmentation-Induced Expansion: Our frequency augmentation expands neighborhoods:

$$\widehat{\mathcal{N}}(S) = \mathcal{N}_{\text{base}}(S) \cup \left\{ x' : \inf_{x \in S} |A(\mathcal{F}(x')) - A(\mathcal{F}(x))|_2 \leq \epsilon \right\}$$

where $\mathcal{N}_{\text{base}}(S)$ is the base augmentation neighborhood. It is worth noting that, unlike most methods, we do not apply base augmentation. The term $\mathcal{N}_{\text{base}}(S)$ is retained in the formulation for theoretical completeness. Our choice of this representation is intended to highlight the expansion gain brought by incorporating frequency augmentation.

The expansion gain becomes:

$$\widehat{c} = c_{\text{base}} + \gamma,$$

$$\gamma = \inf_{S:P(S)\leq a} \frac{P(\widehat{\mathcal{N}}(S) \setminus \mathcal{N}_{\text{base}}(S))}{P(S)}$$

Substituting into Theorem 1, the error becomes:

$$\text{Err}(G) \leq \underbrace{\frac{2}{(c_{\text{base}} + \gamma) - 1}\left(\text{Err}(G_{pl}) - \sum_{k=1}^{K} \Delta_k\right)}_{\text{Reduced by both terms}}$$

$$+ \underbrace{\frac{2(c_{\text{base}} + \gamma)}{(c_{\text{base}} + \gamma) - 1}}_{\text{Reduced by } O(\gamma/c^2)}\mu$$

Notation Summary

- c: Expansion factor (Def. 1).
  Class-wise connectivity metric: Larger c implies stronger neighborhood propagation of local consistency to global predictions.
- μ: Inter-class separation probability (Def. 2).
  Robustness measure: Probability of different classes having overlapping neighborhoods under perturbations, lower μ indicates clearer decision boundaries.
- $\mathcal{R}_B$: Consistency regularizer (Thm. 1).
  Stability term: Penalizes prediction inconsistency between original inputs and their augmented variants.
- Δ_k: Error reduction in k-th sub-domain.
  Specialization gain: Reduced pseudo-label noise in sub-domain k due to frequency-based partitioning.
- γ: Expansion gain from frequency augmentation.
  Neighborhood enhancement: Additional expansion capability measured as the relative increase of augmented neighborhoods.

TABLE II: Classification error rate (↓) on CIFAR-10-C, CIFAR-100-C, and ImageNet-C (IN-C) under **Mixed Distribution Shifts**.

Baseline & Methods	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brig.	Contr.	Elast.	Pixel	JPEG	**Avg.**
CIFAR-10-C (WRN-28)	72.3	65.7	72.9	46.9	54.3	34.8	42.0	25.1	41.3	26.0	9.3	46.7	**26.6**	58.4	30.3	43.5
TBN	45.5	42.8	59.7	34.2	44.3	29.8	32.0	19.8	21.1	21.5	9.3	27.9	33.1	55.5	30.8	33.8
TENT (ICLR 21')	73.5	70.1	81.4	31.6	60.3	29.6	28.5	30.8	35.3	25.7	13.6	44.2	32.6	70.2	34.9	44.1
ETA (ICML 22')	36.2	33.3	52.3	22.9	38.9	22.4	20.5	19.5	19.7	20.4	11.3	35.4	**26.6**	38.8	25.1	28.2
AdaContrast (CVPR 22')	36.7	34.3	48.8	**18.2**	39.1	21.1	17.7	18.6	18.3	16.8	**9.0**	17.4	27.7	44.8	**24.9**	26.2
CoTTA (CVPR 22')	38.7	36.0	56.1	36.0	**36.8**	32.3	31.0	19.9	**17.6**	27.2	11.7	52.6	30.5	35.8	25.7	32.5
SAR (ICLR 23')	45.5	42.7	59.6	34.1	44.3	29.7	31.9	19.8	21.1	21.5	9.3	27.8	33.0	55.4	30.8	33.8
RoTTA (CVPR 23')	60.0	55.5	70.0	23.8	44.1	20.7	21.3	20.2	22.7	**16.0**	9.4	22.7	27.0	58.6	29.2	33.4
RDumb (NeurIPS 23')	34.9	32.3	49.4	23.3	38.2	23.3	20.7	19.9	19.3	20.7	11.2	29.3	26.7	41.5	25.2	27.7
DeYO (ICLR 24')	45.8	42.3	65.7	21.3	41.8	25.1	19.5	21.1	19.6	19.2	12.3	21.8	28.5	39.3	28.0	30.1
UnMix-TNS (ICLR 24')	50.0	44.4	44.3	34.4	48.2	32.7	30.0	35.5	35.9	47.5	28.1	38.7	43.9	40.0	43.3	39.8
FreDA (ours)	**23.1**	**22.2**	**32.2**	18.7	41.6	**18.8**	**16.8**	**17.9**	19.9	16.9	9.8	**13.2**	29.1	**35.4**	28.6	**22.9**
CIFAR-100-C (ResNeXt-29)	73.0	68.0	39.4	29.3	54.1	30.8	28.8	39.5	45.8	50.3	29.5	55.1	37.2	74.7	41.2	46.4
TBN	62.7	60.7	43.1	35.5	50.3	35.7	34.4	39.9	51.5	**27.5**	45.5	42.3	72.8	46.4	45.8	45.8
TENT (ICLR 21')	95.6	95.2	89.2	72.8	82.9	74.4	72.3	78.0	79.7	84.7	71.0	88.5	77.8	96.8	78.7	82.5
ETA (ICML 22')	42.6	40.3	**34.1**	30.3	42.4	32.0	29.4	35.6	35.8	44.1	30.2	41.8	**36.9**	38.9	40.9	37.0
AdaContrast (CVPR 22')	54.5	51.5	37.6	30.7	45.4	32.1	30.3	36.9	36.5	45.3	28.0	42.7	38.2	75.4	41.7	41.8
CoTTA (CVPR 22')	54.4	52.7	49.8	36.0	45.8	36.7	33.9	38.9	35.8	52.0	30.4	60.9	40.2	38.0	41.1	43.1
SAR (ICLR 23')	75.8	72.7	41.1	**29.2**	45.2	31.1	28.9	36.7	37.7	43.9	29.3	41.8	37.1	89.2	42.4	45.5
RoTTA (CVPR 23')	65.0	62.3	39.3	33.4	50.0	34.2	32.6	36.6	36.5	45.0	**26.4**	41.6	40.6	89.5	48.5	45.4
RDumb (NeurIPS 23')	42.3	40.0	**34.1**	30.5	42.4	31.9	29.5	35.7	35.9	43.6	30.4	41.9	36.9	38.1	40.5	36.9
DeYO (ICLR 24')	57.2	53.4	38.8	34.7	47.3	37.3	34.1	40.8	40.5	50.6	33.3	45.8	41.5	94.5	45.7	46.4
UnMix-TNS (ICLR 24')	65.8	64.1	46.4	37.5	51.7	36.0	36.4	38.5	39.4	51.1	29.3	42.8	43.2	67.8	49.4	46.6
FreDA (ours)	**34.8**	**34.7**	36.6	29.4	**41.2**	**29.9**	**28.4**	**33.8**	**33.7**	41.1	29.8	**34.9**	**36.9**	**37.1**	**38.7**	**34.7**
IN-C (ResNet-50)	97.8	97.1	98.2	81.7	89.8	85.2	77.9	83.5	77.1	75.9	41.3	94.5	82.5	79.3	68.6	82.0
TBN	92.8	91.1	92.5	87.8	90.2	87.2	82.2	82.2	82.0	79.8	48.0	92.5	83.5	75.6	70.4	82.5
TENT (ICLR 21')	99.2	98.7	99.0	90.5	95.1	90.5	84.6	86.6	84.0	86.5	46.7	98.1	86.1	77.7	72.9	86.4
ETA (ICML 22')	90.7	89.2	90.5	77.0	**80.6**	74.0	68.9	72.4	70.3	64.6	43.9	93.4	**69.2**	**52.3**	**55.9**	72.9
AdaContrast (CVPR 22')	96.2	95.5	96.2	93.2	96.4	96.3	90.5	92.7	91.9	92.4	50.8	97.0	96.6	89.7	87.1	90.8
CoTTA (CVPR 22')	89.1	86.6	88.5	80.9	87.2	81.1	75.8	73.3	75.2	70.5	41.6	85.0	78.1	65.6	61.6	76.0
SAR (ICLR 23')	98.4	97.3	98.0	84.0	87.3	82.6	77.2	77.5	76.1	72.5	43.1	96.0	78.3	61.8	60.4	79.4
RoTTA (CVPR 23')	89.4	88.6	89.3	83.4	89.1	86.2	80.0	78.9	76.9	74.2	**37.4**	89.6	79.5	69.0	59.6	78.1
RDumb (NeurIPS 23')	89.0	87.6	88.6	78.1	82.3	75.2	70.1	73.0	71.0	65.1	43.9	92.6	70.7	53.7	56.3	73.1
DeYO (ICLR 24')	99.5	99.2	99.5	89.5	95.0	83.9	78.8	75.0	87.8	79.2	47.3	99.2	92.4	59.0	60.4	83.0
UnMix-TNS (ICLR 24')	91.7	92.8	91.7	92.3	93.4	91.5	84.8	86.3	84.1	85.0	62.0	96.5	88.6	81.7	77.3	86.7
FreDA (ours)	**72.4**	**74.0**	**71.4**	**76.5**	82.3	**72.1**	**64.1**	**64.4**	**64.8**	**59.1**	43.7	**79.7**	71.0	54.2	58.6	**67.2**

VII. Experiments

A. Datasets and Experimental Setup

1) Datasets: To provide a comprehensive evaluation of TTA deployment, we test models over multiple datasets under three different scenarios:

- Common Image Corruptions: We evaluate models on CIFAR-10-C, CIFAR-100-C, and ImageNet-C [14] with 10, 100 and 1000 classes, respectively. These benchmarks are designed to assess the model robustness against various corruptions. Each dataset consists of 15 distinct corruptions across five severity levels, resulting in 150,000 at each severity for CIFAR-10-C/100-C, and 750,000 for ImageNet-C.
- Natural Domain Shifts: We extend evaluation to DomainNet126 [35], which presents natural shifts across four domains (Real, Clipart, Painting, Sketch) encompassing 126 classes as a subset of the larger DomainNet dataset.
- Medical Application: Models are further evaluated on Camelyon17 [36], comprising over 450,000 histopathological patches from lymph node sections for binary classification of normal and tumor tissue, with data originating from five distinct healthcare centers.

For corruption datasets, the model is pretrained on the clean dataset and the 15 corruptions are randomly mixed as the target distribution. We leverage the highest severity = 5 in all the experiments. In DomainNet126 and Camelyon17, one subdomain is selected as the source, and the others serve as mixed target distributions. All reported results are averaged over runs with fixed seeds (0, 1, and 2).

2) Baselines: We compare FreDA with 10 models, including TBN [31], TENT [3], CoTTA [5], ETA [6], SAR [7], AdaContrast [4], RoTTA [25], RDumb [9], DeYO [10], and UnMix-TNS [28]. TBN [31] re-estimates batch normalization statistics from test data. TENT [3] minimizes prediction entropy to optimize batch normalization. CoTTA [5] addresses long-term test-time adaptation in changing environments. ETA [6] and SAR [7] exclude unreliable and redundant samples during optimization. AdaContrast [4] utilizes contrastive learning to refine pseudo-labels and improve feature learning. RoTTA [25] presents a robust batch normalization scheme with a memory bank for category-balanced estimation. RDumb [9] leverages weighted entropy and periodically resets the model to its pretrained state to prevent collapse. DeYO [10] quantifies the impact of object-destructive transformations for sample selection and weighting. UnMix-TNS [28] introduces a test-time normalization layer for non-i.i.d. environments by decomposing BN statistics. For fair comparisons, we conduct experiments using the open source online TTA repository [37][1], which provides codes and configurations of state-of-the-art TTA methods.

3) Pretrained Models: We utilize models from RobustBench [38], including WildResNet-28 [39] for CIFAR-10-C and ResNeXt-29 [40] for CIFAR-100-C, both pretrained by [41]. For ImageNet-C, the pretrained ResNet-50 [42] is obtained from `torchvision`. For DomainNet126, pretrained ResNet-50 is sourced from AdaContrast [4], while for Camelyon17, we train a DenseNet-121 [43] from scratch to 100

[1] https://github.com/mariodoebler/test-time-adaptation

TABLE III: Classification error rate (↓) on DomainNet126 and Camelyon17 under **Mixed Distribution Shifts**.

	DomainNet126					Camelyon17					
Methods	Real	Painting	Clipart	Sketch	**Avg.**	C1	C2	C3	C4	C5	**Avg.**
Source	45.2	41.6	49.5	45.3	45.4	21.6	43.6	52.5	47.4	47.6	42.5
TBN	45.5	39.9	45.9	37.5	42.2	26.5	38.5	31.7	**39.4**	32.8	33.8
TENT (ICLR 21’)	42.2	37.8	44.7	37.5	40.6	44.7	50.5	49.9	49.1	48.6	48.6
ETA (ICML 22’)	41.1	37.3	43.4	36.4	39.5	47.4	52.5	47.9	49.9	39.2	47.4
SAR (ICLR 23’)	43.2	38.5	44.8	37.0	40.9	26.5	38.5	31.7	**39.4**	32.8	33.8
DeYO (ICLR 24’)	40.9	36.4	43.6	36.9	39.4	50.4	50.3	48.8	51.7	50.5	50.4
FreDA (ours)	**40.2**	**36.1**	**40.0**	**33.6**	**37.5**	**18.6**	**24.7**	**24.8**	40.5	**30.8**	**27.9**

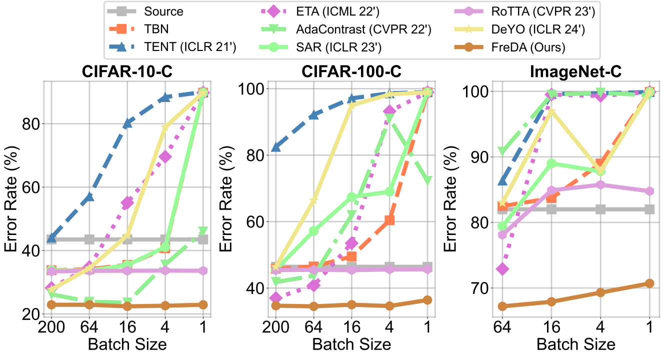


Fig. 5: Averaged classification error rate (↓) on CIFAR-10-C/100-C and ImageNet-C with various batch size under **Mixed Domains**.

epochs with other training specifications outlined in the Wilds benchmark [44].

4) Hyperparameter Configuration: The batch size is set to 200, 64, 128 and 32 for CIFAR-10/100-C, ImageNet-C, DomainNet126 and Camelyon17 following the previous methods. The SGD optimizer is used with learning rates adjusted to 0.01, 0.0001, 0.001 and 0.00005, respectively. The learning rate is proportionally decreased in the experiment studying the effect of batch size. The Kmeans Size is 512, Clutser Number is 4, Communication Interval is 10 across all the tasks. The perturbation magnitude α is fixed to 0.1 and the coefficient λ in loss function is fixed to 0.5. Two threshold in Eq. 5 is set to the same value for corruption datasets and DomainNet126 following ETA [6]. While for Camelyon17, the class diversity related threshold is adjusted to 0.9 empirically.

B. Main Results

1) FreDA improves across diverse distribution shifts: Our method consistently attains the lowest error rates across all evaluated datasets (see TABLE V and III). Notably, on the Camelyon17 dataset, FreDA reduced the error rate to 27.9%, outperforming the next best method by 5.9%. This significant improvement is particularly noTABLE where other approaches falter if compared with no training (TBN), meaning they struggle to adapt to the complex medical imaging data. By effectively handling high variability and intricate patterns in the data, FreDA maintains superior accuracy and adaptability.

2) FreDA remains stable under various batch size: To simulate deployment with constrained batch sizes, we evaluate models under both varying batch sizes and mixed distributions.

TABLE IV: Ablation study of FreDA.

DT	SS	SA	C10	C100	IN
			44.1	82.5	86.4
✓			24.8	54.2	81.2
	✓		29.6	37.5	71.0
		✓	39.4	71.7	92.9
✓	✓		24.3	36.3	69.4
	✓	✓	27.7	36.2	**65.9**
✓		✓	24.4	50.2	77.7
✓	✓	✓	**22.9**	**34.7**	67.2

In Figure 5, we present the results on CIFAR-10-C, CIFAR-100-C, and ImageNet-C using batch sizes ranging from 200 (64) down to 1. Unlike other methods that significantly degrade as batch size decreases – for example the error rate of DeYO increases from 27.7% to 89.8% when batch size drops from 200 to 1 on CIFAR-10-C – FreDA consistently maintains strong performance. This stability demonstrates FreDA’s robustness, making it highly suiTABLE for real-world applications where large batches is not always feasible.

3) FreDA enhances adaptation via synergistic designs: This section validates our designs by ablating three key modules – Decentralized Training (DT), Sample Selection (SS), and Sample Augmentation (SA). The baseline here leverages only the entropy loss. From TABLE VI, we have the following observations: **1)** Implementing decentralized training and sample selection alone results in substantial improvements, reducing error rates dramatically across all datasets. **2)** Sample augmentation alone has the possibility to increase error rates, suggesting that although this approach introduces useful variability, it may introduce unexpected noise under the absence of proper selection or decentralized training. **3)** The combined approach delivers the best performance across all datasets, showing the synergistic effect of our different designs.

C. More analysis and Discussion

1) Performance with Transformer Backbone: In addition to evaluating our model on commonly compared CNN backbones, we further assess its performance under transformer-based architecture, specifically using the ViT-Base backbone. Here, we report results on the ImageNet-C benchmark using ViTBase-LN [45] (see TABLE V), where the pretrained weights are obtained from `torchvision`. Importantly, all experimental configurations are kept consistent with those used in the CNN-based experiments. As shown, our method continues to deliver strong performance, demonstrating its robustness and adaptability across different backbone architectures.

TABLE V: Classification error rate (↓) on ImageNet-C under **Mixed Distribution Shifts** using ViT-Base backbone.

Baseline & Methods	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brig.	Contr.	Elast.	Pixel	JPEG	**Avg.**
IN-C (VitBase-LN)	65.8	67.3	65.3	68.8	74.4	64.3	66.6	56.8	45.2	48.6	29.2	81.8	57.1	60.8	50.2	60.2
TENT (ICLR 21')	60.6	60.4	59.6	63.6	67.8	57.1	61.2	55.0	48.8	47.4	28.6	66.7	53.9	50.4	44.4	55.0
ETA (ICML 22')	59.3	57.8	57.9	58.8	62.8	52.5	58.2	51.0	46.4	44.2	28.8	58.3	51.1	46.9	41.9	51.7
AdaContrast (CVPR 22')	64.8	63.4	63.3	72.8	76.6	73.7	74.6	67.7	48.0	89.6	30.2	93.2	60.8	57.3	46.3	65.5
CoTTA (CVPR 22')	89.4	92.0	88.9	93.6	92.6	90.6	86.5	94.9	88.2	86.6	75.8	96.5	85.7	93.5	84.6	89.3
SAR (ICLR 23')	58.9	57.6	57.6	59.4	63.6	53.0	58.5	52.3	47.1	45.4	28.3	61.6	51.4	47.4	42.0	52.3
RoTTA (CVPR 23')	64.4	65.6	63.7	67.6	71.3	59.8	64.1	52.7	43.5	48.6	**27.9**	78.5	54.3	60.4	50.1	58.2
RDumb (NeurIPS 23')	59.7	58.5	58.5	60.0	64.1	54.0	59.0	52.0	46.7	44.5	28.6	61.2	51.9	48.3	42.6	52.6
DeYO (ICLR 24')	60.0	58.6	58.8	58.8	62.4	61.9	**50.9**	46.7	51.9	45.2	29.7	55.7	51.6	45.8	42.8	52.1
FreDA (ours)	**55.9**	**53.7**	**55.0**	**58.0**	**57.9**	**50.9**	57.4	**45.5**	**42.9**	**43.9**	29.5	**51.7**	**47.8**	**41.6**	**40.7**	**48.8**

TABLE VI: Sensitivity analysis on different datasets.

CLUSTER_NUM	2	**4**	8	16
CIFAR10-C	23.0	22.9	23.2	24.7
CIFAR100-C	34.8	34.7	34.7	35.6
IN-C (ResNet)	68.6	67.2	67.1	70.5
IN-C (ViT)	50.3	48.8	49.9	50.0
KMEANS_SIZE	256	**512**	1024	2048
CIFAR10-C	23.0	22.9	23.0	22.9
CIFAR100-C	34.6	34.7	34.8	34.8
IN-C (ResNet)	69.0	67.2	67.6	67.0
IN-C (ViT)	49.0	48.8	48.7	48.8
COMM_INTERVAL	1	**10**	100	1000
CIFAR10-C	22.6	22.9	22.6	22.0
CIFAR100-C	34.7	34.7	34.9	43.2
IN-C (ResNet)	67.1	67.2	67.2	67.4
IN-C (ViT)	48.4	48.8	48.8	48.7
PERT_MAGNITUDE	0.0	**0.1**	0.2	0.3
CIFAR10-C	24.3	22.9	22.5	22.2
CIFAR100-C	36.3	34.7	34.9	34.9
IN-C (ResNet)	69.3	67.2	67.0	66.9
IN-C (ViT)	49.6	48.8	48.5	48.6

TABLE VII: Classification error rate (↓) on CIFAR-10-C (C10), CIFAR-100-C (C100), and ImageNet-C (IN) using ResNet-50 & ViT-Base backbones under **Continual Setting**, averaged over 15 corruptions.

Methods	**C10**	**C100**	**IN(ResNet)**	**IN(ViT)**
Source	43.5	46.5	82.0	60.2
TBN	20.4	35.4	68.6	-
TENT (ICLR 21')	20.0	62.2	62.6	54.5
ETA (ICML 22')	17.9	32.2	60.2	49.8
AdaContrast (CVPR 22')	18.5	33.5	65.5	57.0
CoTTA (CVPR 22')	**16.5**	32.8	63.1	77.0
SAR (ICLR 23')	20.4	**32.0**	61.9	51.7
RoTTA (CVPR 23')	19.3	34.8	67.3	58.3
RDumb (NeurIPS 23')	17.8	34.1	90.6	50.2
DeYO (ICLR 24')	87.0	98.1	90.6	94.3
UnMix-TNS (ICLR 24')	24.9	32.7	75.4	-
FreDA (ours)	19.5	32.5	**60.2**	**47.9**

2) Sensitivity Study: We investigate the impact of the key hyperparameters: `CLUSTER_NUM`, `KMEANS_SIZE`, `COMM_INTERVAL`, and `PERT_MAGNITUDE`. From TABLE VI, we have the following observations: **1)** The choice of `CLUSTER_NUM` influences model performance, especially on more complex datasets. While performance remains relatively stable even with a small number of clusters (e.g., two clusters), increasing the number of clusters beyond four tends to lead to slight performance degradation, particularly on datasets like ImageNet-C. This suggests that while more clusters can improve adaptation capacity, excessively increasing the number can introduce overfitting and reduce generalization. Therefore, a moderate number of clusters, around 4, appears to strike a good balance between adaptation flexibility and maintaining model robustness. **2)** Varying `KMEANS_SIZE` from 256 to 2048 results in stable performance across all datasets, indicating that our method is robust to changes in cluster sizes. **3)** Our approach shows general robustness to communication frequency (varying `COMM_INTERVAL` from 1 to 1,000). On simpler datasets such as CIFAR10-C, infrequent communication (e.g., interval $f = 1000$) performs best, likely due to effective independent learning. In contrast, for complex datasets, more frequent communication (e.g., $f = 1$) improves performance, likely by mitigating divergence among local branches and ensuring model consistency. **4)** Adjusting the perturbation magnitude (`PERT_MAGNITUDE`) has a noticeable effect, particularly on simpler datasets. Increasing perturbation improves performance, suggesting that spectral augmentation contributes to better generalization. However, for more complex datasets such as CIFAR100-C, higher perturbation levels lead to slight decline in performance, likely due to noise disrupting the alignment of finer features. A perturbation magnitude of 0.1 appears to strike an effective balance, offering the benefits of augmentation without negatively impacting feature alignment.

3) Performance under continual settings: Although our method is specifically designed for mixed domain scenarios, we also evaluated its performance under the conventional continual test-time adaptation [3], [5] setting to assess its robustness in different contexts. In this setting, the model adapts online to a sequence of test domains without explicit knowledge of domain shifts, with only one distribution shift occurring at a time and not reappearing. Without adjusting any parameters, our method demonstrated competitive performance compared to current state-of-the-art approaches. Notably, while UnMix-TNS effectively addresses non-i.i.d. issues (dependent sampling at the class level), it is less effective under i.i.d. conditions. Our results suggest that the proposed FreDA not only excels in its intended mixed domain scenarios but also generalizes effectively to standard continual adaptation tasks, providing a robust solution across various distributional challenges.

VIII. Conclusion and Future Work

This paper advances Test-Time Adaptation (TTA) by tackling the real-world complexities of heterogeneous data streams. Our decentralized approach precisely manages diverse data shifts, improving model adaptation in varied settings. By integrating Fourier-based augmentation, we expand the range of confident samples for each distribution shift, further boosting model performance. The experimental results underscore the efficacy of FreDA, highlighting its potential to influence the field and guide future research in adapting to dynamic and diverse data shifts. While FreDA addresses a critical challenge in handling heterogeneous data streams, providing a solid pipeline for this issue, there are still avenues for further enhancement. Our current aggregation approach, which averages models based on cluster counts, has been effective in solving the problem at hand. However, exploring alternative strategies – such as weighting models by the divergence between clusters – might lead to incremental improvements. Additionally, refining the sample selection process from a original sample-level focus to a more granular patch-level could extend FreDA's applicability to tasks such as segmentation, further enhancing its versatility in real-world scenarios.

Appendix

A. Adaptation Scenarios

1) Mixed Domains: : In this scenario, the model processes a long sequence of test samples where each sample $x_i \sim p_{e_i}(x)$ is independently drawn from a randomly selected target domain $\mathcal{D}_{e_i} \in \{\mathcal{D}_{t_1}, \mathcal{D}_{t_2}, \ldots, \mathcal{D}_{t_N}\}$ and a randomly selected class c_i among classes $\{1, 2, \ldots, C\}$. The sequence is represented as:

$$\left\{ x_1^{\mathcal{D}_{e_1}, c_1},\ x_2^{\mathcal{D}_{e_2}, c_2},\ \ \ldots,\ x_k^{\mathcal{D}_{e_k}, c_k} \right\},$$

where each target domain index $e_i \in \{1, 2, \ldots, N\}$ and class number $c_i \in \{1, 2, \ldots, C\}$ are independently and randomly selected for each sample x_i.

2) Continual Domain Adaptation: : In this setting, the model adapts online to a sequentially presented series of test domains, where each domain shift occurs only once and does not reappear. The sequence progresses through distinct target domains in a fixed order $D_1 \rightarrow D_2 \rightarrow \cdots \rightarrow D_N$, with samples from each domain appearing contiguously. The class labels within each domain are independently and randomly selected. The sequence is structured as:

$$\underbrace{\{(x_1^{D_1, c_1}), (x_2^{D_1, c_2}), \ldots, (x_{l_1}^{D_1, c_{l_1}})\}}_{\text{Samples from domain } D_1}$$
$$\rightarrow \underbrace{\{(x_{l_1+1}^{D_2, c_{l_1+1}}), \ldots, (x_{l_1+l_2}^{D_2, c_{l_1+l_2}})\}}_{\text{Samples from domain } D_2}$$
$$\rightarrow \cdots \rightarrow \underbrace{\{(x_{l_1+\cdots+l_{N-1}+1}^{D_N, c_{l_1+\cdots+l_{N-1}+1}}), \ldots, (x_k^{D_N, c_k})\}}_{\text{Samples from domain } D_N}$$

where:

- D_i denotes the i-th target domain in the fixed sequence, with $i \in \{1, 2, \ldots, N\}$.
- $c_m \in \{1, 2, \ldots, C\}$ is the randomly selected class label for the m-th sample, independent of domain transitions.

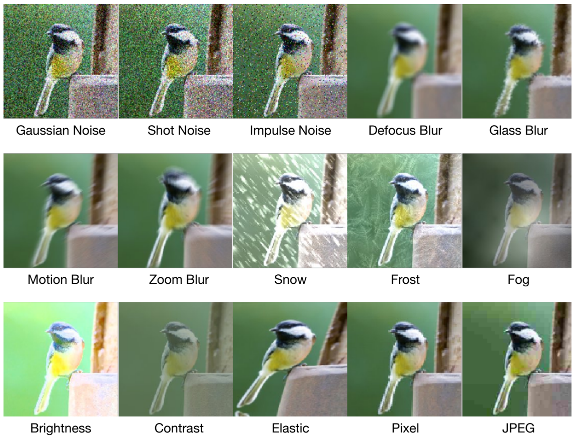


Fig. 6: Examples from ImageNet-C under common image corruptions. The images showcase a range of corruption types (e.g., noise, blur, and weather distortions) at varying severity levels.

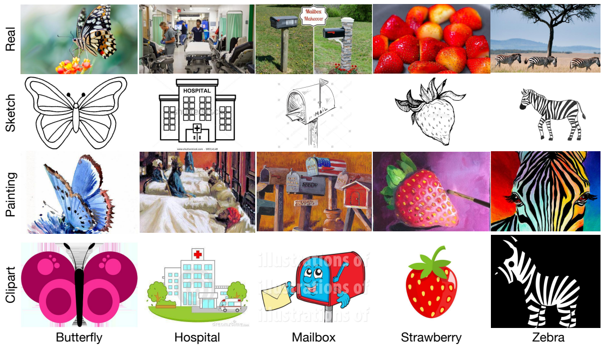


Fig. 7: Samples from DomainNet126 across four subdomains (Real, Sketch, Painting, Clipart). These visualizations reflect the stylistic and perceptual variations inherent in each domain.

B. Dataset Visualization

To further illustrate the characteristics of the datasets used in our evaluation, we present visualizations of the data distribution across different corruption types (Fig. 6), natural domain shifts (Fig. 7), and medical centers (Fig. 2). These figures highlight the diverse challenges that our models face in each evaluation scenario, providing insight into the complexity of the test conditions.

C. Relation to Frequency Domain Learning

Recent advances highlight frequency-based techniques as powerful tools for domain transfer. In domain generalization, frequency analysis has revealed critical insights into model robustness and learning dynamics [46]–[53]. For domain adaptation, interpolating image amplitude spectra across styles has proven effective in reducing domain gaps and preventing overfitting to low-level statistics [54]–[57]. Motivated by these advancements, we discover that frequency information inherently captures domain characteristics, making it a valuable medium for decoupling mixed target domains – an aspect largely unexplored in prior work. On this basis, we propose

FreDA, the first framework addressing TTA under mixed domain shifts in Fourier space, with a decentralized adaptation and perturbation mechanism.

D. Relation to Multi-Target Domain Adaptation

TTA under mixed distribution shifts shares similarities with multi-target unsupervised domain adaptation (MT-UDA) [11], [12], [58], [59], yet diverges critically in complexity: while MT-UDA assumes static, predefined target domains and leverages labeled source data for explicit domain alignment, TTA operates with no access to source data and must adapt to dynamic, unpredicTABLE target streams in an online manner. This eliminates direct source-target discrepancy computation and demand robust incremental adaptation rather than offline multi-domain optimization. These differences highlight the need for novel methodologies tailored specifically to TTA, going beyond the solutions developed for MT-UDA.

E. Relation to Decentralized, Federated, Distributed Learning

This work also intersects with decentralized, federated, and distributed learning due to splitting data batches into disjoint subsets and applying decentralized model adaptation. First, while decentralized learning focuses on non-i.i.d. data that is naturally distributed across multiple nodes [60], our approach starts with a centralized batch of target samples. By proactively splitting this data into disjoint subsets, we expose latent non-i.i.d. characteristics, enabling the effective use of decentralized learning techniques. Second, federated learning considers data privacy and model collaborations within decentralized learning [61]. In our case, as target samples are mixed in a batch, data privacy is not a concern. However, similar to federated learning, our approach also involves weight aggregation over subnetworks to enhance their base models before next batch adaptation. Third, distributed learning aims to improve training efficiency on large-scale datasets by partitioning data for synchronized training [62]. In contrast, our method operates in a real-time fine-tuning context with limited data at one time, hence scalability is less of a concern.

References

[1] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2200–2207.

[2] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1180–1189.

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

[4] D. Chen, D. Wang, T. Darrell, and S. Ebrahimi, "Contrastive test-time adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 295–305.

[5] Q. Wang, O. Fink, L. Van Gool, and D. Dai, "Continual test-time domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7201–7211.

[6] S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao, and M. Tan, "Efficient test-time model adaptation without forgetting," in *International Conference on Machine Learning*. PMLR, 2022, pp. 16 888–16 905.

[7] S. Niu, J. Wu, Y. Zhang, Z. Wen, Y. Chen, P. Zhao, and M. Tan, "Towards stable test-time adaptation in dynamic wild world," in *International Conference on Learning Representations*, 2023.

[8] Z. Su, J. Guo, K. Yao, X. Yang, Q. Wang, and K. Huang, "Unraveling batch normalization for realistic test-time adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 13, 2024, pp. 15 136–15 144.

[9] O. Press, S. Schneider, M. Kümmerer, and M. Bethge, "Rdumb: A simple approach that questions our progress in continual test-time adaptation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[10] J. Lee, D. Jung, S. Lee, J. Park, J. Shin, U. Hwang, and S. Yoon, "Entropy is not enough for test-time adaptation: From the perspective of disentangled factors," in *International Conference on Learning Representations*, 2024.

[11] B. Gholami, P. Sahu, O. Rudovic, K. Bousmalis, and V. Pavlovic, "Unsupervised multi-target domain adaptation: An information theoretic approach," *IEEE Transactions on Image Processing*, vol. 29, pp. 3993–4002, 2020.

[12] T. Isobe, X. Jia, S. Chen, J. He, Y. Shi, J. Liu, H. Lu, and S. Wang, "Multi-target domain adaptation with collaborative consistency learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8187–8196.

[13] K. Yao, Z. Tan, Z. Su, X. Yang, J. Sun, and K. Huang, "Scmix: Stochastic compound mixing for open compound domain adaptation in semantic segmentation," *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[14] D. Hendrycks and T. G. Dietterich, "Benchmarking neural network robustness to common corruptions and surface variations," *arXiv preprint arXiv:1807.01697*, 2018.

[15] Z. Su, J. Guo, X. Yang, Q. Wang, F. Coenen, and K. Huang, "Navigating distribution shifts in medical image analysis: A survey," *arXiv preprint arXiv:2411.05824*, 2024.

[16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[17] L. Zhang and X. Gao, "Transfer adaptation learning: A decade survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 23–44, 2022.

[18] S. Yang, K. Yu, F. Cao, L. Liu, H. Wang, and J. Li, "Learning causal representations for robust domain adaptation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 2750–2764, 2021.

[19] Z. Z. Darban, Y. Yang, G. I. Webb, C. C. Aggarwal, Q. Wen, S. Pan, and M. Salehi, "Dacad: Domain adaptation contrastive learning for anomaly detection in multivariate time series," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[20] C. Li, Y. Song, and Y.-H. Shao, "Domain adaptation via learning using statistical invariant," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[21] C. Zhang, F. Nie, and R. Wang, "Anchor guided unsupervised domain adaptation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 3, pp. 1079–1090, 2025.

[22] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9229–9248.

[23] Y. Liu, P. Kothari, B. Van Delft, B. Bellot-Gurlet, T. Mordan, and A. Alahi, "Ttt++: When does self-supervised test-time training fail or thrive?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 808–21 820, 2021.

[24] X. Yang, Y. Wang, J. Chen, W. Fan, X. Zhao, E. Zhu, X. Liu, and D. Lian, "Dual test-time training for out-of-distribution recommender system," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[25] L. Yuan, B. Xie, and S. Li, "Robust test-time adaptation in dynamic scenarios," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 922–15 932.

[26] T. Gong, J. Jeong, T. Kim, Y. Kim, J. Shin, and S.-J. Lee, "Note: Robust continual test-time adaptation against temporal correlation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 253–27 266, 2022.

[27] B. Zhao, C. Chen, and S.-T. Xia, "Delta: Degradation-free fully test-time adaptation," in *International Conference on Learning Representations*, 2023.

[28] D. Tomar, G. Vray, J.-P. Thiran, and B. Bozorgtabar, "Un-mixing test-time normalization statistics: Combatting label temporal correlation," in *International Conference on Learning Representations*, 2024.

[29] R. A. Marsden, M. Döbler, and B. Yang, "Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 2555–2565.

[30] 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 *International Conference on Machine Learning*. PMLR, 2023, pp. 42 574–42 588.

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

[32] Y. Zhang, X. Wang, K. Jin, K. Yuan, Z. Zhang, L. Wang, R. Jin, and T. Tan, “Adanpc: Exploring non-parametric classifier for test-time adaptation,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 41 647–41 676.

[33] A. Karmanov, D. Guan, S. Lu, A. El Saddik, and E. Xing, “Efficient test-time adaptation of vision-language models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 162–14 171.

[34] C. Wei, K. Shen, Y. Chen, and T. Ma, “Theoretical analysis of self-training with deep networks on unlabeled data,” *arXiv preprint arXiv:2010.03622*, 2020.

[35] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, “Semi-supervised domain adaptation via minimax entropy,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8050–8058.

[36] P. Bandi, O. Geessink, Q. Manson, M. Van Dijk, M. Balkenhol, M. Hermsen, B. E. Bejnordi, B. Lee, K. Paeng, A. Zhong *et al.*, “From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 550–560, 2018.

[37] M. Döbler, R. A. Marsden, and B. Yang, “Robust mean teacher for continual and gradual test-time adaptation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7704–7714.

[38] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, “Robustbench: a standardized adversarial robustness benchmark,” in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. [Online]. Available: https://openreview.net/forum?id=SSKZPJCt7B

[39] S. Zagoruyko and N. Komodakis, “Wide residual networks,” *arXiv preprint arXiv:1605.07146*, 2016.

[40] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.

[41] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, “Augmix: A simple data processing method to improve robustness and uncertainty,” in *International Conference on Learning Representations*, 2020.

[42] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[43] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[44] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao *et al.*, “Wilds: A benchmark of in-the-wild distribution shifts,” in *International conference on machine learning*. PMLR, 2021, pp. 5637–5664.

[45] A. Dosovitskiy, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[46] H. Wang, X. Wu, Z. Huang, and E. P. Xing, “High-frequency component helps explain the generalization of convolutional neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8684–8694.

[47] Z. J. Xu, “Understanding training and generalization in deep learning by fourier analysis,” *arXiv preprint arXiv:1808.04295*, 2018.

[48] Z.-Q. J. Xu, Y. Zhang, and Y. Xiao, “Training behavior of deep neural network in frequency domain,” in *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I 26*. Springer, 2019, pp. 264–274.

[49] D. Yin, R. Gontijo Lopes, J. Shlens, E. D. Cubuk, and J. Gilmer, “A fourier perspective on model robustness in computer vision,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[50] R. Soklaski, M. Yee, and T. Tsiligkaridis, “Fourier-based augmentations for improved robustness and uncertainty calibration,” *arXiv preprint arXiv:2202.12412*, 2022.

[51] P. Vaish, S. Wang, and N. Strisciuglio, “Fourier-basis functions to bridge augmentation gap: Rethinking frequency augmentation in image classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 763–17 772.

[52] C. Liu, W. Xiang, Y. He, H. Xue, S. Zheng, and H. Su, “Improving model generalization by on-manifold adversarial augmentation in the frequency domain,” *arXiv preprint arXiv:2302.14302*, 2023.

[53] Y. Long, Q. Zhang, B. Zeng, L. Gao, X. Liu, J. Zhang, and J. Song, “Frequency domain model augmentation for adversarial attack,” in *European conference on computer vision*. Springer, 2022, pp. 549–566.

[54] Y. Yang and S. Soatto, “Fda: Fourier domain adaptation for semantic segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4085–4095.

[55] C. Yang, X. Guo, Z. Chen, and Y. Yuan, “Source free domain adaptation for medical image segmentation with fourier style mining,” *Medical Image Analysis*, vol. 79, p. 102457, 2022.

[56] Q. Xu, R. Zhang, Y. Zhang, Y. Wang, and Q. Tian, “A fourier-based framework for domain generalization,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 383–14 392.

[57] Q. Xu, R. Zhang, Z. Fan, Y. Wang, Y.-Y. Wu, and Y. Zhang, “Fourier-based augmentation with applications to domain generalization,” *Pattern Recognition*, vol. 139, p. 109474, 2023.

[58] Z. Liu, Z. Miao, X. Pan, X. Zhan, D. Lin, S. X. Yu, and B. Gong, “Open compound domain adaptation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 406–12 415.

[59] T. Feng, H. Shi, X. Liu, W. Feng, L. Wan, Y. Zhou, and D. Lin, “Open compound domain adaptation with object style compensation for semantic segmentation,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[60] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, “The non-iid data quagmire of decentralized machine learning,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 4387–4398.

[61] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[62] R. McDonald, K. Hall, and G. Mann, “Distributed training strategies for the structured perceptron,” in *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, 2010, pp. 456–464.