UniTTA: Unified Benchmark and Versatile Framework Towards Realistic Test-Time Adaptation

Anonymous Authors[1]

Abstract

We propose a **Uni**fied **T**est-**T**ime **A**daptation (**UniTTA**) benchmark, which is comprehensive and widely applicable. Alongside this benchmark, we propose a versatile UniTTA framework, which includes a Balanced Domain Normalization (BDN) layer and a COrrelated Feature Adaptation (COFA) method–designed to mitigate distribution gaps in domain and class, respectively. Extensive experiments demonstrate that our framework excels within the UniTTA benchmark and achieves state-of-the-art performance on average.

1. Introduction

Recent studies have extended TTA to more realistic scenarios, proposing various methods to address challenges such as continual domain shifts (Wang et al., 2022), mixed domains (Marsden et al., 2024; Tomar et al., 2024), and temporally correlated (Boudiaf et al., 2022; Gong et al., 2022; Yuan et al., 2023) or imbalanced class distributions (Su et al., 2024). However, many of the current methods have only been evaluated in specific scenarios and lack a unified and comprehensive benchmark for performance assessment.

To address this issue, we propose a **Uni**fied **T**est-**T**ime **A**daptation (**UniTTA**) benchmark that is both comprehensive and widely applicable. We present a novel method for constructing test data of various scenarios using a defined Markov state transition matrix. The UniTTA benchmark can assist researchers in evaluating their methods in a more comprehensive and realistic manner, facilitating the development of versatile and robust TTA methods. Moreover, it also provides a evaluating benchmark for practitioners to select the most suitable TTA method for their specific scenarios. To obtain a versatile and robust TTA method, we need to simultaneously address domain and class distribution shifts. This poses two primary challenges: potential domain correlation and imbalance leading to inaccurate domain-wise statistics, and class correlation and imbalance further biasing domain-wise statistics towards majority classes.

In this work, we simultaneously tackle both challenges by proposing a novel Balanced Domain Normalization (BDN) layer. Our primary insight is to unify both domain-aware and class-aware normalization. We compute the statistics for each class within each domain and then average across classes to obtain balanced domain-wise statistics, mitigating the impact of class imbalance on domain-wise statistics. During prediction, we select the corresponding statistics based on the current sample's domain, effectively addressing domain correlation and imbalance. Moreover, to address potential temporal correlation of class, we leverage the correlation characteristic by referencing the feature of the previous sample, resulting in an effective and efficient method named COFA (COrrelated Feature Adaptation), without requiring any modifications to model parameters.

2. Benchmark

2.1. Existing Realistic TTA Settings

The realistic TTA settings can be divided into two categories: domain setting and class setting, as shown in Tab. 1. For the class setting, real-world data streams are typically highly correlated, which means that data categories do not change abruptly. Given these scenarios, we can classify the factors in existing realistic TTA settings into two categories: Temporal Correlation and Imbalance. Therefore, a more general realistic TTA setting should consider different combinations of these factors to better simulate real-world scenarios. Based on this analysis, a natural question arises: *how can we generate such a data stream?*

2.2. UniTTA Benchmark

We propose a new UniTTA benchmark, based on a Markov state transition matrix from a novel local perspective. In the following discussion, we consider the temporal correlation and imbalance of domains and classes as two independent factors. Specifically, *the Markov state can represent either the domain or the class of the data*. Our key idea is to generate data that satisfies temporal correlation by controlling the probability of samples transitioning to themselves. While this method might appear to neglect the issue of data imbalance, we have discovered that by properly configuring the Markov state transition matrix, *we can effectively address both temporal correlation and imbalance simultaneously*.

Table 1. Comparison of the proposed UniTTA benchmark with existing realistic TTA settings.

Realistic TTA Setting	Method	Domain Setting		Class Setting	
		Temporal Correlation	Imbalance	Temporal Correlation	Imbalance
Correlated TTA (Boudiaf et al., 2022)	LAME	N/A (Single)	N/A (Single)	Correlated	Imbalanced/Balanced
Continual TTA (Wang et al., 2022)	CoTTA	Continual	Balanced	i.i.d.	Balanced
Practical TTA (Yuan et al., 2023)	RoTTA	Continual	Balanced	Correlated	Balanced
GLI-TTA (Su et al., 2024)	TRIBE	Continual	Balanced	Correlated	Imbalanced/Balanced
Mixed Domain (Marsden et al., 2024)	ROID	i.i.d.	Balanced	i.i.d	Balanced
UniTTA Benchmark	UniTTA	Continual/Correlated/i.i.d.	Imbalanced/Balanced	Continual/Correlated/i.i.d.	Imbalanced/Balanced

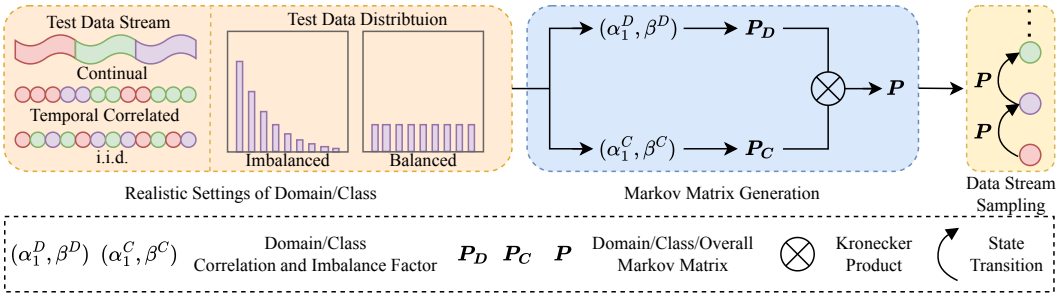

Figure 1. Data generation process for the UniTTA benchmark. Continual TTA describes a scenario in which the domain remains consistent over an extended period before shifting to a new domain, which exemplifies an extreme case of correlation settings. We consider the domain and class as two independent attributes, each associated with its own Markov matrix.

First, we define a simple uniformly leaving Markov state transition matrix $\boldsymbol{P}$, where each element P_{ij} represents the probability of transitioning from state i to state j. Intuitively, this transition matrix implies that the probability of transitioning from any state to any other state is uniform.

Definition 1 (Uniformly Leaving Markov Matrix)**.** A **Uniformly Leaving Markov Matrix (ULMM)** is a transition matrix in a Markov chain where each non-diagonal entry P_{ij}, representing the transition probability from state i to state j (where $i \neq j$), is identical across all states j. Specifically, the matrix is defined as:

$$\boldsymbol{P_{ij}} = \begin{cases} \frac{1-P_{ii}}{n-1} & \text{if } i \neq j \\ P_{ii} & \text{if } i = j \end{cases} \tag{1}$$

Based on the above definition, the ULMM can be characterized by a single vector $\boldsymbol{\alpha}$, where $\alpha_i = P_{ii}$. This matrix has n degrees of freedom. By adjusting $\alpha_i \in \left[\frac{1}{n}, 1\right]$, we can generate data with varying levels of temporal correlation. Therefore, we refer to $\boldsymbol{\alpha}$ as the (temporal) correlation vector and α_i as the (temporal) correlation factor.

A key question we address is whether the state distribution of data sampled from a ULMM satisfies the criteria for imbalance. According to Markov Chain theory, this distribution corresponds to the stationary distribution of the matrix, as stated in the following proposition:

Proposition 1 (Stationary Distribution)**.** *For a Uniformly Leaving Markov Matrix with diagonal elements $\boldsymbol{\alpha}$ where $\alpha_i = P_{ii}$ for all i, there exists a unique stationary distribution $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_n)$. This distribution satisfies the following relationship: $(1-\alpha_1)\pi_1 = (1-\alpha_2)\pi_2 = \cdots = (1-\alpha_n)\pi_n$.*

To ensure that the sampled data follows a long-tail distribution (assuming, without loss of generality, that $\frac{\pi_1}{\pi_2} = \cdots = \frac{\pi_{n-1}}{\pi_n} \geq 1$, where $\frac{\pi_1}{\pi_n} = \beta$ is the imbalance factor), the configurations of $\boldsymbol{\alpha}$ are described by the following corollary:

Corollary 1 (Temporal Correlation and Imbalance)**.** *If the category distribution of data sampled based on a Uniformly Mixing Markov Matrix follows a long-tailed (power law) distribution characterized by an imbalance factor $\beta \geq 1$. Under these conditions, $\boldsymbol{\alpha}$ are constrained such that:*

$$\frac{1-\alpha_1}{1-\alpha_n} = \frac{1}{\beta}, \quad \textit{and} \quad \frac{1-\alpha_1}{1-\alpha_2} = \cdots = \left(\frac{1}{\beta}\right)^{\frac{1}{n-1}}. \tag{2}$$

Additionally, if the distribution exhibits temporal correlation, which implies that $\alpha_1, \alpha_2, \ldots, \alpha_n > \frac{1}{n}$, then the following inequality holds: $(1-\alpha_1)\beta < \frac{n-1}{n}$, and $\alpha_1 < 1$.

In summary, as shown in Fig. 1, generating data that satisfies both temporal correlation and imbalance requires tuning two parameters of the ULMM. *Specifically, it is sufficient to set the (maximum temporal) correlation factor $\alpha_1 \in [1/n, 1]$ and the imbalance factor $\beta \in [1, \infty)$ to satisfy inequality in Corollary 1.* The remaining α_i can then be determined. We then combine the domain and class ULMMs using the Kronecker product to obtain a final ULMM for sampling, where the (domain, class) pair is treated as a new state.

3. UniTTA Framework

3.1. Overview

As illustrated in the Fig. 2, the UniTTA framework utilizes a progressive prediction strategy through three forward passes: **Forward 1:** In the absence of prior domain and class information, we perform a forward pass using global statistics to obtain initial pseudo-labels. **Forward 2:** With class labels available, we conduct a second forward pass, updating both class and global statistics. At a specified BDN layer (as a hyper-parameter), we also predict the domain based on

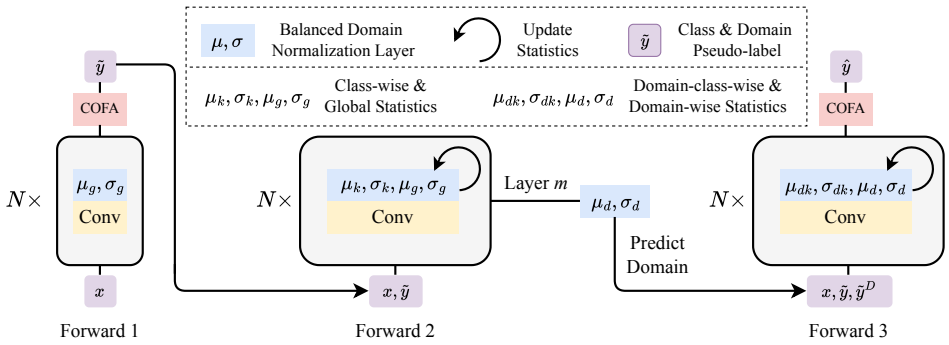

Figure 2. The overall architecture of the UniTTA framework. The original model's BN layers are replaced by BDN layers, and the linear classifier is equipped with the COFA method. The UniTTA framework sequentially predicts the class label and domain label in the m-th BDN layer through three forward passes, ultimately providing the final prediction.

domain statistics. **Forward 3:** Finally, with both class and domain labels, we perform a forward pass using domain statistics for the final prediction, updating both domain-class and domain statistics.

3.2. Balanced Domain Normalization

The core idea of Balanced Domain Normalization (BDN) is to implement a domain-aware normalization in an unsupervised manner. To counteract the bias caused by imbalanced class data, which skews domain statistics towards the majority classes, we suggest calculating both domain-specific and class-specific statistics. By averaging these statistics, we can remove the class bias and obtain more accurate domain statistics. For each sample, we calculate the instance statistics (Ulyanov et al., 2016) ($\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i^2$) of the feature map, which are essential for domain assignment, expansion, and updating the statistics.

Domain assignment (prediction) and expansion. First, all domain statistics, including those generated by expansions, are initialized using the corresponding batch normalization (BN) statistics of the original pretrained model ($\boldsymbol{\mu}_{\text{ori}}$, $\boldsymbol{\sigma}^2_{\text{ori}}$). Initially, the number of domains is set to one. Next, domain assignment and the decision to expand the domain are performed at a specific layer, which is the only hyper-parameter in our method. Specifically, we calculate the Kullback-Leibler (KL) divergence between the instance statistics of each sample and the domain statistics, assuming they follow a normal distribution. If the KL divergence of the sample to all domain statistics is *greater than that to the original domain statistics, the sample is considered to belong to a new domain*, necessitating domain expansion during the Forward 3. This condition is satisfied when:

$$\begin{aligned} &\min_d D^S_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2) \,||\, \mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\sigma}_d^2)) \\ &> D^S_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2) \,||\, \mathcal{N}(\boldsymbol{\mu}_{\text{ori}}, \boldsymbol{\sigma}^2_{\text{ori}})), \end{aligned} \tag{3}$$

where D^S_{KL} is the symmetric KL divergence. Otherwise, the sample is assigned to the domain with the minimum KL divergence:

$$\hat{y}_i^D = \operatorname*{argmin}_d D^S_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2) \,||\, \mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\sigma}_d^2)). \tag{4}$$

Domain-class statistics update. Based on the domain assignment, we update the domain-class statistics ($\boldsymbol{\mu}_{dk}$, $\boldsymbol{\sigma}_{dk}$) and domain statistics ($\boldsymbol{\mu}_d$, $\boldsymbol{\sigma}_d$) using the instance statistics $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i^2$. The class statistics ($\boldsymbol{\mu}_k$, $\boldsymbol{\sigma}_k$) and global statistics ($\boldsymbol{\mu}_g$, $\boldsymbol{\sigma}_g$) can be considered as the domain-class statistics and domain statistics for a single domain, respectively. Various updating methods can be applied independently of our core method. We utilize the commonly applied Exponential Moving Average (EMA) to update the domain-class statistics $\boldsymbol{\mu}_{dk}$ and $\boldsymbol{\sigma}_{dk}$. Specifically, we adopt the EMA update from Balanced BN (Su et al., 2024) without modification. For detailed update rules, please refer to App. D.

3.3. Correlated Feature Adaptation

The COFA method leverages the correlation characteristics of data to enhance prediction accuracy by utilizing the information of the previous sample when predicting the current sample. Implementing this method is straightforward, requiring only the storage of feature from the previous sample. Specifically, the classifier with COFA is defined as follows:

$$\boldsymbol{p}_i^{\text{COFA}} = \text{softmax}\left(\frac{\boldsymbol{w}^T(\boldsymbol{z}_i + \boldsymbol{z}_{i-1})}{2} + b\right), \tag{5}$$

where $\boldsymbol{z}_i$ is the feature of the i-th sample, and $\boldsymbol{w}$ and b are the weight and bias of the original classifier of the pretrained model, respectively. However, direct implementation of COFA results in a marked performance decrease under i.i.d. conditions. To address this, we propose a confidence filtering strategy to combine the predictions of the COFA and the original classifier, as follows:

$$\boldsymbol{p}_i = \begin{cases} \boldsymbol{p}_i^{\text{COFA}}, & \text{if } \max(\boldsymbol{p}_i^{\text{COFA}}) > \max(\boldsymbol{p}_i^{\text{single}}) \\ \boldsymbol{p}_i^{\text{single}}, & \text{otherwise} \end{cases} \tag{6}$$

where $\boldsymbol{p}_i^{\text{single}} = \text{softmax}(\boldsymbol{w}^T\boldsymbol{z}_i + b)$. By filtering out low-confidence predictions, COFA balances performance in both i.i.d. and correlation conditions.

Table 2. **Average error (%) on ImageNet-C within the UniTTA benchmark.** $(\{i, n, 1\}, \{1, u\})$ denotes correlation and imbalance settings, where $\{i, n, 1\}$ represent i.i.d., correlated and continual, respectively, and $\{1, u\}$ represent balance and imbalance, respectively. Corresponding setting denotes the existing setting and method as shown in Tab. 1.

Class setting	i.i.d. and balanced (i,1)		correlated and balanced (n,1)					correlated and imbalanced (n,u)					
Domain setting	(1,1)	(i,1)	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	
Corresponding setting	CoTTA	ROID	RoTTA	–	–	–	–	TRIBE	–	–	–	–	Avg.
TENT (Wang et al., 2021)	70.58	91.88	98.72	99.31	99.53	99.12	99.32	97.50	99.22	99.13	97.03	98.86	95.85
ROID (Marsden et al., 2024)	**60.67**	79.18	98.51	99.71	99.84	99.52	99.61	91.76	99.77	99.57	98.15	99.37	93.80
NOTE (Gong et al., 2022)	91.62	88.18	93.67	95.27	96.82	95.00	95.81	92.49	95.93	95.41	88.93	95.05	93.68
CoTTA (Wang et al., 2022)	66.87	80.67	95.13	96.80	97.33	96.22	96.33	89.70	95.20	94.50	92.11	93.71	91.22
TRIBE (Su et al., 2024)	75.85	84.78	89.78	92.62	96.54	95.19	95.99	88.72	92.85	93.71	89.37	94.05	90.79
BN (Nado et al., 2020)	69.33	82.87	93.79	95.08	95.15	95.10	95.01	88.40	92.24	92.25	91.31	91.84	90.20
UnMIX-TNS (Tomar et al., 2024)	79.64	85.55	79.74	84.42	82.67	84.57	82.91	78.67	83.28	82.34	85.04	82.38	82.60
TEST	81.99	82.05	81.92	82.10	81.66	81.96	81.74	81.60	81.21	81.42	81.20	81.52	81.70
RoTTA (Yuan et al., 2023)	67.77	79.91	71.72	80.54	79.65	80.30	79.63	68.74	78.26	77.94	79.78	78.36	76.88
LAME (Boudiaf et al., 2022)	82.55	82.26	74.48	72.21	71.77	73.52	73.13	75.70	73.44	73.54	74.38	74.39	75.12
UniTTA	78.07	**78.00**	**70.25**	**66.83**	**66.42**	**68.29**	**68.05**	**72.02**	**65.68**	**66.87**	**68.48**	**67.58**	**69.71** (-5.41)

4. Experiments

In this section, we mainly present the main results on our proposed UniTTA benchmark in Sec. 4.1. For detailed information on the experimental setup and more results, please refer to App. E and App. G.

4.1. Main Results

To better simulate real-world scenarios, we exclude the continual setting for classes, as it is rare for all samples from a single class to appear consecutively in practice. We present the results for 12 of these settings in the main paper, encompassing both existing and the most challenging scenarios. *Additional results for all methods and components of all 24 settings for all three datasets are available in App. G.* We can compare the robustness of different methods across various datasets and settings in Tab. 2. Our method outperforms the others on all datasets across most settings, consistently achieving superior performance, particularly in more realistic scenarios.

4.2. Ablation Study

We conduct an ablation study across various settings and datasets to evaluate the impact of different components, benchmarking them against similar methods as shown in Tab. 3 and Tab. 4. This section presents the overall results, while detailed results are provided in App. G.

(a) *Effectiveness of different components.* We first investigate the impact of different components on model performance across all settings and datasets. The results in Tab. 3 demonstrate the effectiveness of our two core components, COFA and BDN. Additionally, applying the confidence filter further enhances model performance.

(b) *Comparison with similar methods.* We compare our two components with both parameter-free method which do not require modifications to model parameters and normalization methods. Our BDN consistently outperforms other normalization methods, including UnMIX-TNS (Tomar et al., 2024) and Balanced BN (Su et al., 2024). Notably, our COFA achieves performance comparable to LAME by leveraging the temporal correlation characteristic (just averaging with the lastest feature).

Table 3. Ablation study of different components. The average of 12 settings are reported on CIFAR10-C, CIFAR100-C, and ImageNet-C.

	C10-C	C100-C	IN-C	Avg.
TEST	42.03	46.42	81.70	56.72
COFA	37.22	37.34	76.38	50.31
BN (Nado et al., 2020)	46.97	68.06	90.20	68.41
BDN	26.64	40.88	77.15	48.22
UniTTA	**20.68**	**32.43**	**69.71**	**40.94**

Table 4. Comparison of our two components with parameter-free and normalization methods.

	C10-C	C100-C	IN-C	Avg.
Parameter-free Method				
LAME (Boudiaf et al., 2022)	40.12	**36.38**	**75.12**	50.74
COFA	**37.22**	37.34	76.38	**50.31**
Normalization Method				
Robust BN (Yuan et al., 2023)	32.34	46.33	85.30	54.66
UnMIX-TNS (Tomar et al., 2024)	30.84	44.75	82.60	52.73
Balanced BN (Su et al., 2024)	30.10	43.83	82.54	52.17
BDN	**26.64**	**40.88**	**77.15**	**48.22**

5. Conclusion

In this work, we propose a unified benchmark, UniTTA, for Test-Time Adaptation. It sets a benchmark for evaluating realistic TTA scenarios and provides a guideline for selecting the most suitable TTA method for specific scenarios. Building on this, we introduce a versatile UniTTA framework consisting of a Balanced Domain Normalization (BDN) layer and a COFA method, which are simple and effective without additional training. Empirical evidence from the UniTTA benchmark demonstrates that our framework excels in various Realistic TTA scenarios and achieves state-of-the-art performance on average.

Impact Statements

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Boudiaf, M., Mueller, R., Ben Ayed, I., and Bertinetto, L. Parameter-free online test-time adaptation. In *CVPR*, 2022.

Brahma, D. and Rai, P. A probabilistic framework for lifelong test-time adaptation. In *CVPR*, 2023.

Gong, T., Jeong, J., Kim, T., Kim, Y., Shin, J., and Lee, S.-J. Note: Robust continual test-time adaptation against temporal correlation. In *NeurIPS*, 2022.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2004.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

Huang, G. and Du, C. The high separation probability assumption for semi-supervised learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52 (12):7561–7573, 2022.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Marsden, R. A., Döbler, M., and Yang, B. Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In *WACV*, 2024.

Mirza, M. J., Micorek, J., Possegger, H., and Bischof, H. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *CVPR*, 2022.

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

Ross, S. M. *Stochastic processes*. John Wiley & Sons, 1995.

Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, 2020.

Su, Y., Xu, X., and Jia, K. Towards real-world test-time adaptation: Tri-net self-training with balanced normalization. In *AAAI*, 2024.

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

Tomar, D., Vray, G., Thiran, J.-P., and Bozorgtabar, B. Unmixing test-time normalization statistics: Combatting label temporal correlation. In *ICLR*, 2024.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Wang, Q., Fink, O., Van Gool, L., and Dai, D. Continual test-time domain adaptation. In *CVPR*, 2022.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

Yuan, L., Xie, B., and Li, S. Robust test-time adaptation in dynamic scenarios. In *CVPR*, 2023.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference*, 2016.

Zou, Y., Zhang, Z., Li, C.-L., Zhang, H., Pfister, T., and Huang, J.-B. Learning instance-specific adaptation for cross-domain segmentation. In *ECCV*, 2022.

A. Related Work

Test-Time Adaptation (TTA) addresses distributional shifts in test data without requiring additional data acquisition or labeling. Sun et al.(Sun et al., 2020) propose an on-the-fly adaptation method using an auxiliary self-supervised task. Subsequent TTA algorithms (Nado et al., 2020; Schneider et al., 2020; Wang et al., 2021) leverage batches of test samples to recalibrate Batch Normalization (BN) layers(Ioffe & Szegedy, 2015) using test data. These studies show that using test batch statistics in BN layers can enhance robustness against distributional shifts. TENT (Wang et al., 2021) refines this approach by adapting a pre-trained model to test data through entropy minimization (Grandvalet & Bengio, 2004), updating a few trainable parameters in BN layers.

Realistic Test-Time Adaptation. Recent studies on Test-Time Adaptation (TTA) have investigated more realistic scenarios, addressing distribution changes in test data. These studies consider factors such as domain distribution shift (Wang et al., 2022; Brahma & Rai, 2023), temporal correlation (Boudiaf et al., 2022; Gong et al., 2022), and combinations of both (Yuan et al., 2023; Marsden et al., 2024; Su et al., 2024; Tomar et al., 2024). A comprehensive comparison of these realistic settings is provided in Sec. 2.1. The methods employed in these studies include self-training (Wang et al., 2022; Yuan et al., 2023; Brahma & Rai, 2023), which integrates semi-supervised self-training techniques (Huang & Du, 2022) to enhance model performance, parameter-free methods (Boudiaf et al., 2022) utilizing Laplacian regularization, and Batch Normalization (BN) recalibration (Gong et al., 2022; Mirza et al., 2022; Zou et al., 2022; Yuan et al., 2023; Tomar et al., 2024; Sun et al., 2020). RoTTA (Yuan et al., 2023) introduces robust BN, estimating global statistics via exponential moving average. TRIBE (Su et al., 2024) proposes a balanced BN (BBN) layer, consisting of multiple category-wise BN layers for unbiased statistic estimation. UnMIX-TNS (Tomar et al., 2024) unmixes correlated batches into K distinct components, each reflecting statistics from similar test inputs. Among these methods, BBN and UnMIX-TNS are the most similar to our work. However, both BBN and UnMIX-TNS consider the influence of category and domain distributions on statistics separately, which significantly limits their applicability. In contrast, our approach simultaneously accounts for both category and domain distributions by introducing a unified BDN layer to address their combined impact on statistics.

B. Proof

Proof of Prop. 1. By the convergence properties of Markov chains (Ross, 1995), a Uniformly Leaving Markov Matrix (ULMM) $\boldsymbol{P}$ has a unique stationary distribution $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)$ which satisfies $\boldsymbol{\pi} = \boldsymbol{\pi}\boldsymbol{P}$. To solve this, we must find the nontrivial solution to the linear equation $(P^T - I)\boldsymbol{\pi} = \mathbf{0}$, where I is the identity matrix and $\boldsymbol{\pi}$ is a column vector. Thus, we have

$$\begin{pmatrix} \alpha_1 - 1 & \frac{1-\alpha_2}{n-1} & \cdots & \frac{1-\alpha_n}{n-1} \\ \frac{1-\alpha_1}{n-1} & \alpha_2 - 1 & \cdots & \frac{1-\alpha_n}{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1-\alpha_1}{n-1} & \frac{1-\alpha_2}{n-1} & \cdots & \alpha_n - 1 \end{pmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \\ \begin{pmatrix} -1 & \frac{1}{n-1} & \cdots & \frac{1}{n-1} \\ \frac{1}{n-1} & -1 & \cdots & \frac{1}{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n-1} & \frac{1}{n-1} & \cdots & -1 \end{pmatrix} \begin{pmatrix} (1-\alpha_1)\pi_1 \\ (1-\alpha_2)\pi_2 \\ \vdots \\ (1-\alpha_n)\pi_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{7}$$

Observing that each row of the coefficient matrix sums to zero, there exists a non-trivial solution $\mathbf{1} = (1, 1, \ldots, 1)$. Hence,

$$(1-\alpha_1)\pi_1 = (1-\alpha_2)\pi_2 = \cdots = (1-\alpha_n)\pi_n \tag{8}$$

is one of the non-trivial solutions. By the uniqueness of the stationary distribution, the proof is complete. □

C. Sampling Time and Discussion of UniTTA Benchmark

C.1. Sampling Time

A practical concern is the time required for the sampling process. Given that the ULMM matrix remains stationary most of the time (except for certain special data points, as previously discussed), we can pre-sample a sequence of transitions for each state before the actual sampling begins. During the sampling process, we can then directly use these pre-sampled sequences which ensures that the ULMM-based data generation method does not result in significantly higher time costs compared to existing methods as shown in Tab. 5.

Table 5. Time of sampling 750k data on ImageNet-C (corrected and imbalanced).

Sampling Method	Time (s)
naive sampling	22.40
+ pre-sampling	2.15

C.2. Discussion

In this section, we discuss the scalability of the UniTTA benchmark. By independently generating domain and class ULMMs, we can create a comprehensive ULMM for sampling. Moreover, the sampling ULMM can be enhanced by considering the relationships between domains and classes. This allows us to construct domain-dependent class ULMMs, where the transition probability of a class depends on the current domain, and vice versa. Additionally, the ULMM can be adapted for various scenarios, such as temporal anti-correlation scenarios, non-uniform scenarios where transition probabilities to other states are unequal, and higher-order Markov Chains, where transition probabilities depend on multiple previous states, not just the current one. In summary, the data generation method defined by the UniTTA benchmark is *highly flexible and can be efficiently extended to meet the requirements of real-world scenarios*.

D. Implementation Details

Before introducing the statistical update rules of BDN, we define a mean notation to simplify the expressions:

$$\overline{F_{c,\cdot,\cdot}} = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} F_{c,h,w} \tag{9}$$

which denotes the average over the omitted dimensions. Using this definition, we can simplify instance statistics as follows:

$$\boldsymbol{\mu}_i = \overline{F_{c,\cdot,\cdot}}, \quad \boldsymbol{\sigma}_i^2 = \overline{(F_{c,\cdot,\cdot} - \boldsymbol{\mu}_i)^2}. \tag{10}$$

We adopt the update rules of Balanced BN from TRIBE (Su et al., 2024) to update the statistics of BDN. For a sample with pseudo-label domain d and class k, the update rules are simplified as follows:

$$\boldsymbol{u}_{dk} \leftarrow (1-\eta)\boldsymbol{u}_{dk} + \eta\overline{F_{c,\cdot,\cdot}} \tag{11}$$

$$\boldsymbol{\sigma}_{dk}^2 \leftarrow (1-\eta)\boldsymbol{\sigma}_{dk}^2 + \eta\overline{(F_{c,\cdot,\cdot} - \boldsymbol{u}_{dk})^2} - \eta^2(\overline{F_{c,\cdot,\cdot}} - \boldsymbol{u}_{dk})^2 \tag{12}$$

$$\boldsymbol{\mu}_d \leftarrow \overline{\boldsymbol{u}_{d\cdot}} \tag{13}$$

$$\boldsymbol{\sigma}_d^2 \leftarrow \overline{\boldsymbol{\sigma}_{d\cdot}^2} + \overline{(\boldsymbol{u}_{d\cdot} - \boldsymbol{\mu}_d)^2} \tag{14}$$

where the momentum coefficient η is set to $5 \times 10^{-4} \times K_C$ following TRIBE and K_C is the number of classes. Specifically, for global statistics, to enhance their robustness, we also follow the approach of TRIBE by incorporating the class-agnostic updating strategy (Robust BN (Yuan et al., 2023)) with a parameter γ.

E. Experimental Setup

We conduct experiments on three test-time adaptation datasets: CIFAR10-C (Hendrycks & Dietterich, 2019), CIFAR100-C (Hendrycks & Dietterich, 2019), and ImageNet-C (Hendrycks & Dietterich, 2019). Each dataset includes 15 different corruptions at 5 levels of severity. We evaluate all methods under the highest corruption severity level, level 5. Following previous works (Wang et al., 2021; 2022; Yuan et al., 2023; Su et al., 2024), we adopt a standard pre-trained WideResNet-28 (Zagoruyko & Komodakis, 2016), ResNeXt-29 (Xie et al., 2017), and ResNet-50 (He et al., 2016) as the backbone networks for CIFAR10-C, CIFAR100-C, and ImageNet-C, respectively. The batch size is set to 64 for CIFAR10-C and CIFAR100-C, and 32 for ImageNet-C. For all comparison methods, we use the original optimizers, learning rate schedules, and hyperparameter settings as described in the respective papers. All experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU.

For our UniTTA framework, mainly following the results of Fig. 5, we set the BDN layer for domain prediction to the block2.layer.0.bn1, stage2.0.bn and layer3.0.bn1 for WideResNet-28, ResNeXt-29, and ResNet-50, respectively. For all settings of the UniTTA benchmark, unless otherwise specified, the correlation factor α_1 of correlation settings for the domain and class is 0.85 and 0.95, respectively. The imbalance factor β for the domain and class is 5 and 10, respectively. The correlation factor α_1 is $1/K$ for the i.i.d. settings, where K is the number of classes or domains. For the balanced settings, the imbalance factor β is 1. For the continual settings, the correlation factor α_1 is 1.

F. More Analysis

Evaluation under more correlation/imbalance factors. Additional experiments are conducted under varying correlation and imbalance factors as shown in Fig. 3. The settings are both correlated and imbalanced in terms of domain and class distribution. The results indicate that our method remains robust across different correlation and imbalance factors.

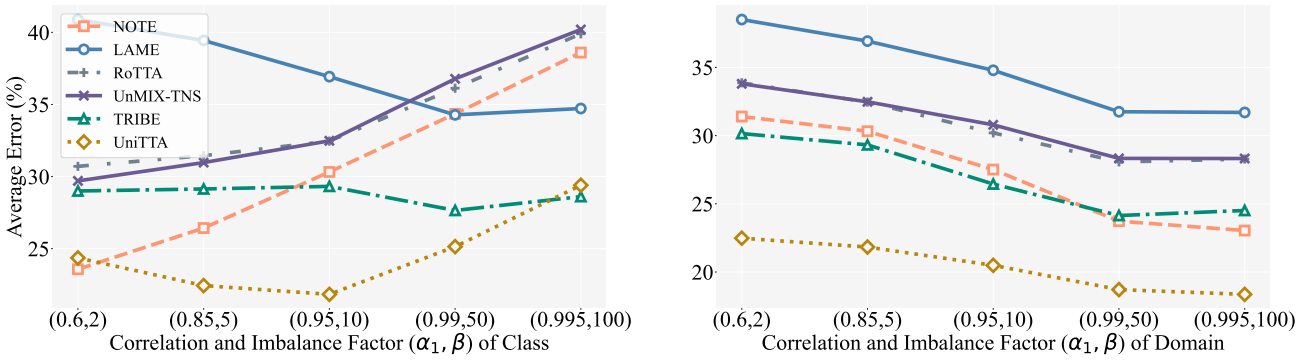

Figure 3. Average error (%) on CIFAR10-C under various correlation and imbalance factors. The default factors for domain and class are (0.85, 5) and (0.95, 10), respectively. In two sets of experiments, we kept either the domain or class factors constant while varying the other.

Hyperparameter Sensitivity. We also conduct experiments to assess the sensitivity to hyperparameters. Fig. 4 shows the performance of several competitive baselines and our method under different batch sizes. Our method's performance remains unaffected by batch size, which can be attributed to the inherent characteristics of the BDN and COFA methods. In contrast, batch-based methods such as LAME and NOTE exhibit significant sensitivity to batch size.

Our framework has only one hyperparameter: the position of the BDN for domain prediction. The results in Fig. 5 show that the performance of BDN is optimal when the first layer of an intermediate block is selected. This also indicates that the network retains more of the original image information in the shallow layers while learning more class-specific features in the deeper layers.

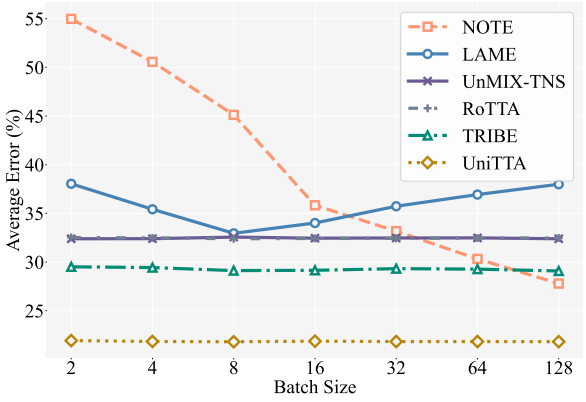

Figure 4. Sensitive analysis of batch size on CIFAR10-C. The default correlation and imbalance factors for domain and class are $(0.85, 5)$ and $(0.95, 10)$, rspectively.

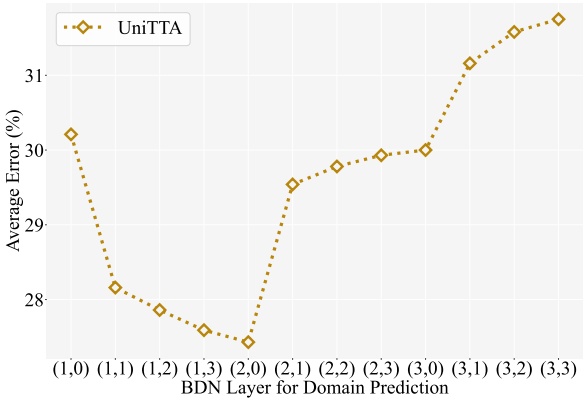

Figure 5. Sensitivity analysis of the BDN layer for domain prediction on CIFAR100-C. The horizontal axis (m, n) indicates the nth layer of the mth block in the network.

Visualization of dynamic domain expansion. We also visualize the domain expansion process in Fig. 6. The process demonstrates that the BDN layer effectively captures the domain information and dynamically expands domains, which is crucial for accurate domain prediction.

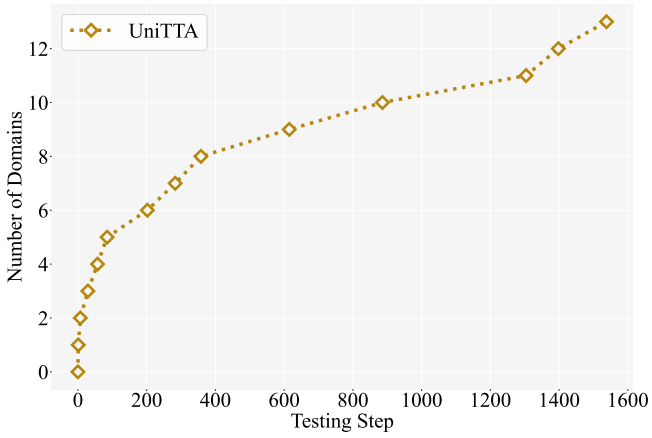

Figure 6. Visualization of dynamic domain expansion on CIFAR10-C. The BDN layer dynamically expands the domains based on the KL divergence of the domain-wise statistics. Only domains with more than 100 samples are counted.

G. Results on More Settings

Table 6. **Average error (%) on CIFAR10-C within the UniTTA benchmark.** $(\{i, n, 1\}, \{1, u\})$ denotes correlation and imbalance settings, where $\{i, n, 1\}$ represent i.i.d., correlated and continual, respectively, and $\{1, u\}$ represent balance and imbalance, respectively. Corresponding setting denotes the existing setting and method as shown in Tab. 1.

Class setting	correlated and balanced (n,1)						correlated and imbalanced (n,u)					
Domain setting	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)
Corresponding setting	RoTTA	–	–	–	–	–	TRIBE	–	–	–	–	–
TENT (Wang et al., 2021)	70.29	83.23	73.16	78.79	69.18	60.13	47.40	59.57	51.48	62.00	51.90	44.89
TEST	43.76	43.52	40.37	43.45	40.68	40.30	42.46	42.83	38.74	42.29	39.39	38.77
LAME (Boudiaf et al., 2022)	41.40	40.48	36.98	40.15	37.49	37.62	40.91	40.64	36.58	40.16	36.93	36.88
ROID (Marsden et al., 2024)	43.79	56.93	53.78	53.55	52.75	42.84	41.35	52.55	48.47	51.43	48.84	40.08
CoTTA (Wang et al., 2022)	53.21	63.93	62.77	61.67	61.21	51.96	41.46	55.18	50.27	53.20	50.42	40.42
BN (Nado et al., 2020)	49.42	57.00	54.82	56.26	54.91	48.47	41.52	50.65	46.52	50.08	47.27	39.94
Robust BN (Yuan et al., 2023)	23.34	35.61	31.99	36.04	32.44	22.01	26.52	38.52	34.09	39.63	35.16	24.78
UnMIX-TNS (Tomar et al., 2024)	24.68	32.99	29.03	32.72	29.15	25.25	27.60	35.81	31.88	36.44	32.48	27.48
Balanced BN (Su et al., 2024)	21.37	34.13	30.28	34.25	30.71	20.04	22.25	34.79	31.02	35.68	31.64	20.54
RoTTA (Yuan et al., 2023)	19.52	36.89	31.49	35.66	31.58	20.51	20.39	36.24	31.67	36.33	32.46	20.53
NOTE (Gong et al., 2022)	31.79	38.52	27.58	32.98	28.49	26.28	34.92	34.79	28.58	33.99	30.32	29.28
TRIBE (Su et al., 2024)	18.54	32.37	28.34	32.57	28.87	17.75	**17.75**	32.60	28.69	32.92	29.32	16.87
COFA(w/o filter)	37.63	31.19	28.33	36.26	32.10	33.70	37.91	32.43	29.05	36.74	32.68	33.83
COFA	38.88	34.95	31.70	37.95	34.25	35.03	37.80	34.88	31.09	37.25	33.41	33.73
BDN (w/o filter)	24.83	28.14	25.42	28.37	25.36	23.31	25.75	30.12	26.89	30.20	27.22	23.87
BDN	22.04	28.97	25.89	28.90	25.96	20.45	22.77	30.57	27.11	30.68	27.44	20.86
UniTTA	**16.40**	**18.53**	**16.19**	**20.09**	**17.20**	**15.34**	17.93	**20.88**	**18.46**	**22.89**	**19.88**	**16.41**

Table 7. **Average error (%) on CIFAR10-C within the UniTTA benchmark.** Continuation of the previous table. "Avg." represents the average error rate across 24 settings.

Class setting	i.i.d. and balanced (i,1)						i.i.d. and imbalanced (i,u)						
Domain setting	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)	
Corresponding setting	CoTTA	ROID	–	–	–	–	–	–	–	–	–	–	Avg.
TENT (Wang et al., 2021)	24.03	59.37	38.58	47.07	37.81	20.88	23.36	48.18	39.40	36.45	32.34	22.03	49.23
TEST	43.46	43.45	40.30	43.52	40.49	40.22	42.46	42.83	38.93	42.52	39.53	38.82	41.38
LAME (Boudiaf et al., 2022)	45.07	44.60	41.35	44.94	41.62	41.68	42.92	42.93	39.02	42.79	39.58	39.15	40.49
ROID (Marsden et al., 2024)	16.92	31.00	27.39	28.03	26.13	**15.71**	34.13	45.68	41.53	43.83	41.20	32.59	40.44
CoTTA (Wang et al., 2022)	**16.68**	33.76	28.75	27.67	25.52	15.99	18.86	35.18	33.01	33.39	35.08	18.64	40.34
BN (Nado et al., 2020)	21.00	34.18	30.23	32.12	29.39	18.78	26.18	38.46	34.04	36.13	34.00	23.78	39.80
Robust BN (Yuan et al., 2023)	20.90	33.81	29.89	34.30	30.17	19.35	26.00	38.25	33.86	38.28	34.64	24.00	30.98
UnMIX-TNS (Tomar et al., 2024)	24.53	32.82	28.80	32.98	28.91	24.85	27.59	35.82	31.74	35.82	32.54	27.50	30.39
Balanced BN (Su et al., 2024)	21.18	33.85	30.03	34.31	30.22	19.60	22.25	34.97	30.92	34.82	31.52	20.31	28.78
RoTTA (Yuan et al., 2023)	17.84	33.45	29.50	33.58	29.73	18.78	18.88	35.62	31.21	35.19	31.79	19.32	28.67
NOTE (Gong et al., 2022)	22.55	**24.48**	**22.33**	**24.06**	**22.35**	21.85	26.39	30.62	**25.87**	**29.37**	**26.48**	24.79	28.28
TRIBE (Su et al., 2024)	18.20	31.90	27.97	32.29	28.04	17.29	**17.77**	32.71	28.19	31.96	28.82	**16.47**	26.18
COFA(w/o filter)	65.98	62.97	61.96	65.45	63.52	64.31	63.74	60.90	59.57	63.22	60.88	61.65	48.17
COFA	47.75	46.71	43.82	47.59	44.64	44.58	46.18	45.41	41.99	46.08	43.15	42.65	40.06
BDN (w/o filter)	24.71	27.46	24.62	27.74	24.49	22.71	25.57	**29.64**	26.04	29.62	26.97	23.38	26.35
BDN	21.22	28.16	25.02	28.36	24.85	19.42	22.53	30.18	26.40	29.92	27.01	20.50	25.63
UniTTA	28.38	31.34	28.44	31.81	28.58	26.25	28.89	32.39	28.99	32.77	30.31	26.54	**23.95** (-2.23)

Table 8. **Average error (%) on CIFAR100-C within the UniTTA benchmark.** $(\{i, n, 1\}, \{1, u\})$ denotes correlation and imbalance settings, where $\{i, n, 1\}$ represent i.i.d., correlated and continual, respectively, and $\{1, u\}$ represent balance and imbalance, respectively. Corresponding setting denotes the existing setting and method as shown in Tab. 1.

Class setting	correlated and balanced (n,1)						correlated and imbalanced (n,u)					
Domain setting	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)
Corresponding setting	RoTTA	–	–	–	–	–	TRIBE	–	–	–	–	–
TENT (Wang et al., 2021)	96.53	97.08	94.26	95.08	89.75	94.91	93.79	93.74	86.80	88.26	83.97	90.80
NOTE (Gong et al., 2022)	79.69	67.67	57.69	59.75	54.37	65.43	71.52	58.86	55.52	57.55	54.25	61.70
BN (Nado et al., 2020)	76.55	79.33	78.55	79.15	79.42	76.22	64.42	69.33	69.68	69.11	68.49	63.69
CoTTA (Wang et al., 2022)	78.26	79.95	78.91	78.89	79.38	76.77	65.68	68.56	69.58	68.07	68.46	63.95
ROID (Marsden et al., 2024)	71.09	77.57	76.80	76.14	76.47	70.56	55.27	63.21	63.88	62.70	63.38	54.83
RoTTA (Yuan et al., 2023)	38.95	53.80	52.30	53.25	52.55	40.44	37.79	54.99	53.89	55.36	53.63	40.34
TEST	46.64	46.66	45.11	47.33	44.89	45.11	47.07	46.86	45.83	47.87	46.05	45.04
Robust BN (Yuan et al., 2023)	40.90	50.09	48.75	51.17	49.13	40.36	39.33	48.50	48.14	49.90	48.48	38.64
UnMIX-TNS (Tomar et al., 2024)	39.12	46.88	45.66	46.92	45.36	40.19	40.19	47.44	46.41	47.55	46.20	41.00
Balanced BN (Su et al., 2024)	36.36	46.47	45.66	47.01	45.16	36.47	36.77	46.67	46.39	47.30	46.40	36.47
TRIBE (Su et al., 2024)	34.69	47.95	43.75	47.89	44.37	35.02	32.74	46.67	43.19	46.35	44.37	32.83
LAME (Boudiaf et al., 2022)	34.07	32.80	30.44	33.19	29.84	31.83	37.44	36.43	34.75	37.08	34.86	35.04
COFA(w/o filter)	32.88	**28.52**	**26.98**	32.69	**28.98**	30.96	36.04	31.89	**30.58**	36.71	33.45	34.05
COFA	35.65	32.66	31.11	35.87	32.65	33.84	37.09	34.21	33.04	37.70	34.97	34.94
BDN (w/o filter)	38.85	44.35	43.52	44.99	43.62	39.62	38.58	44.37	44.38	44.74	44.53	38.48
BDN	36.19	43.46	42.70	44.10	42.86	36.37	36.03	43.50	43.41	43.63	43.34	35.91
UniTTA	**24.49**	28.99	28.57	**31.85**	29.53	**25.11**	**25.81**	**30.96**	30.95	**32.87**	**32.16**	**26.26**

Table 9. **Average error (%) on CIFAR100-C within the UniTTA benchmark.** Continuation of the previous table. "Avg." represents the average error rate across 24 settings.

Class setting	i.i.d. and balanced (i,1)						i.i.d. and imbalanced (i,u)						
Domain setting	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)	
Corresponding setting	CoTTA	ROID	–	–	–	–	–	–	–	–	–	–	Avg.
TENT (Wang et al., 2021)	81.06	91.05	83.37	88.59	79.70	63.18	76.04	73.53	58.98	53.84	50.11	60.56	81.87
NOTE (Gong et al., 2022)	65.96	63.07	56.56	63.47	56.10	57.57	67.54	56.62	54.39	55.59	52.80	57.30	60.46
BN (Nado et al., 2020)	36.20	46.48	44.93	45.04	44.27	35.34	37.36	47.70	46.64	46.45	44.78	36.53	57.74
CoTTA (Wang et al., 2022)	32.74	43.02	42.47	41.22	41.91	32.61	33.47	44.37	45.01	43.60	43.65	33.56	56.42
ROID (Marsden et al., 2024)	**29.91**	**36.84**	**36.65**	**36.81**	**36.71**	**29.90**	31.89	**38.71**	**39.31**	**38.84**	**38.42**	**31.70**	51.57
RoTTA (Yuan et al., 2023)	33.46	46.54	46.63	47.28	46.46	35.41	34.00	51.43	51.30	53.71	50.63	36.07	46.68
TEST	46.35	46.43	44.55	46.72	44.53	44.53	46.94	46.80	45.84	47.43	44.48	44.88	46.00
Robust BN (Yuan et al., 2023)	35.56	45.99	44.45	46.64	44.56	35.18	36.73	46.97	46.32	47.91	44.99	36.29	44.37
UnMIX-TNS (Tomar et al., 2024)	38.94	46.32	44.66	46.61	44.75	39.78	39.96	47.16	46.23	47.58	44.94	40.78	44.19
Balanced BN (Su et al., 2024)	35.84	45.94	44.38	46.38	44.43	35.62	36.32	46.50	45.82	46.98	44.71	35.99	42.75
TRIBE (Su et al., 2024)	33.10	45.73	42.99	46.84	43.38	32.99	**31.71**	45.28	43.01	46.60	41.65	31.82	41.04
LAME (Boudiaf et al., 2022)	48.21	47.47	45.59	47.90	45.58	46.34	48.23	47.34	46.35	48.00	45.06	46.00	40.41
COFA(w/o filter)	70.82	69.50	68.33	70.65	69.22	69.52	70.60	69.43	68.70	70.66	68.41	69.45	50.79
COFA	51.64	51.50	49.83	52.04	49.98	50.02	52.18	51.71	50.73	52.65	49.99	50.29	42.76
BDN (w/o filter)	37.82	43.70	42.17	43.56	41.95	37.42	37.77	44.20	43.44	45.02	42.33	37.86	41.97
BDN	34.65	41.92	40.55	42.29	40.41	34.15	35.06	42.43	42.10	43.54	41.34	34.58	40.19
UniTTA	44.17	48.86	47.72	49.36	47.78	44.08	44.14	49.18	49.05	50.33	47.57	43.72	**38.06** (-2.35)

Table 10. **Average error (%) on ImageNet-C within the UniTTA benchmark.** $(\{i, n, 1\}, \{1, u\})$ denotes correlated and imbalance settings, where $\{i, n, 1\}$ represent i.i.d., correlated and continual, respectively, and $\{1, u\}$ represent balance and imbalance, respectively. Corresponding setting denotes the existing setting and method as shown in Tab. 1.

Class setting	correlated and balanced (n,1)						correlated and imbalanced (n,u)					
Domain setting	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)
Corresponding setting	RoTTA	–	–	–	–	–	TRIBE	–	–	–	–	–
NOTE (Gong et al., 2022)	93.67	95.27	96.82	95.00	95.81	88.75	92.49	95.93	95.41	88.93	95.05	85.85
TENT (Wang et al., 2021)	98.72	99.31	99.53	99.12	99.32	97.69	97.50	99.22	99.13	97.03	98.86	94.71
TRIBE (Su et al., 2024)	89.78	92.62	96.54	95.19	95.99	78.04	88.72	92.85	93.71	89.37	94.05	69.34
ROID (Marsden et al., 2024)	98.51	99.71	99.84	99.52	99.61	97.35	91.76	99.77	99.57	98.15	99.37	91.75
BN (Nado et al., 2020)	93.79	95.08	95.15	95.10	95.01	93.76	88.40	92.24	92.25	91.31	91.84	88.34
CoTTA (Wang et al., 2022)	95.13	96.80	97.33	96.22	96.33	94.59	89.70	95.20	94.50	92.11	93.71	89.04
Robust BN (Yuan et al., 2023)	80.76	89.58	89.58	91.74	90.40	81.08	74.69	87.16	87.31	89.24	88.46	75.82
UnMIX-TNS (Tomar et al., 2024)	79.74	84.42	82.67	84.57	82.91	82.08	78.67	83.28	82.34	85.04	82.38	81.52
TEST	81.92	82.10	81.66	81.96	81.74	82.07	81.60	81.21	81.42	81.20	81.52	81.95
Balanced BN (Su et al., 2024)	76.63	87.03	86.54	88.87	87.23	77.24	**71.19**	84.38	84.41	86.22	85.35	72.27
LAME (Boudiaf et al., 2022)	74.48	72.21	71.77	73.52	73.13	74.69	75.70	73.44	73.54	74.38	74.39	76.25
RoTTA (Yuan et al., 2023)	71.72	80.54	79.65	80.30	79.63	73.59	68.74	78.26	77.94	79.78	78.36	72.47
COFA(w/o filter)	75.37	70.61	69.75	74.42	73.22	75.30	76.32	71.07	70.57	75.06	74.56	76.67
COFA	76.62	73.86	73.51	76.17	75.41	76.97	76.82	73.28	73.59	75.82	75.77	77.29
BDN (w/o filter)	77.80	76.37	76.03	76.88	76.38	79.21	76.69	75.13	75.74	75.97	75.99	78.48
BDN	76.69	79.48	79.32	79.82	79.28	77.68	72.87	77.83	78.21	77.89	78.12	74.16
UniTTA	**70.25**	**66.83**	**66.42**	**68.29**	**68.05**	**72.39**	72.02	**65.68**	**66.87**	**68.48**	**67.58**	**71.70**

Table 11. **Average error (%) on ImageNet-C within the UniTTA benchmark.** Continuation of the previous table. "Avg." represents the average error rate across 24 settings.

Class setting	i.i.d. and balanced (i,1)						i.i.d. and imbalanced (i,u)						
Domain setting	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)	(1,1)	(i,1)	(i,u)	(n,1)	(n,u)	(1,u)	
Corresponding setting	CoTTA	ROID	–	–	–	–	–	–	–	–	–	–	Avg.
NOTE (Gong et al., 2022)	91.62	88.18	83.53	86.89	87.55	85.78	88.90	94.75	94.02	91.06	94.64	83.18	91.21
TENT (Wang et al., 2021)	70.58	91.88	82.37	80.13	85.00	64.92	68.21	97.29	96.06	87.28	94.29	62.02	90.01
TRIBE (Su et al., 2024)	75.85	84.78	83.59	84.61	83.61	63.98	79.88	85.48	87.38	87.02	84.86	62.18	84.98
ROID (Marsden et al., 2024)	**60.67**	79.18	83.83	77.54	78.46	**62.25**	**57.70**	76.61	76.82	73.52	76.20	**58.95**	84.86
BN (Nado et al., 2020)	69.33	82.87	83.22	79.40	80.45	69.44	67.57	81.79	81.58	77.73	79.33	68.19	84.71
CoTTA (Wang et al., 2022)	66.87	80.67	82.07	**76.28**	76.81	66.13	64.31	79.42	78.28	**72.69**	75.74	63.42	83.89
Robust BN (Yuan et al., 2023)	69.81	84.90	85.16	87.35	85.64	70.55	68.37	84.37	84.17	86.22	85.65	69.36	82.81
UnMIX-TNS (Tomar et al., 2024)	79.64	85.55	88.41	86.68	84.48	81.81	78.15	82.91	81.91	83.01	82.12	81.08	82.72
TEST	81.99	82.05	83.46	82.78	82.15	82.14	80.93	81.00	81.15	80.69	81.40	81.17	81.72
Balanced BN (Su et al., 2024)	69.31	83.35	84.71	85.40	83.60	69.89	67.28	82.07	82.17	83.51	82.90	68.46	80.42
LAME (Boudiaf et al., 2022)	82.55	82.26	83.83	83.05	82.41	82.68	81.42	81.13	81.30	80.98	81.65	81.75	78.02
RoTTA (Yuan et al., 2023)	67.77	79.91	81.22	81.11	79.81	71.98	66.16	75.81	75.71	77.65	76.36	71.26	76.07
COFA(w/o filter)	91.83	89.69	90.93	91.95	91.15	92.01	91.32	88.99	89.19	90.75	90.93	91.64	82.22
COFA	82.99	82.77	84.07	83.52	83.26	83.24	82.03	81.81	81.95	81.79	82.46	82.19	79.05
BDN (w/o filter)	77.09	76.64	**80.28**	77.31	76.84	78.09	75.79	**74.93**	**75.20**	75.72	75.88	77.33	76.74
BDN	68.62	**77.64**	80.83	76.68	**76.54**	68.96	66.64	75.76	76.02	74.66	**75.56**	67.43	75.69
UniTTA	78.07	78.00	80.89	78.32	77.94	79.28	76.76	75.96	76.71	76.45	76.91	78.30	**73.26** (-2.81)