Towards Robust Continual Test-Time Adaptation via Neighbor Filtration

Taki Hasan Rafi
Hanyang University
Seoul, South Korea
takihr@hanyang.ac.kr

Amit Agarwal
Liverpool John Moores University
United Kingdom
amit.pinaki@gmail.com

Hitesh L. Patel
New York University
New York, USA
hitesh.patel945@gmail.com

Dong-Kyu Chae*
Hanyang University
Seoul, South Korea
dongkyu@hanyang.ac.kr

ABSTRACT

Test-Time Adaptation (TTA) aims to adapt an unseen target domain utilizing the unlabeled target data using a pre-trained source model. Continual TTA is a more challenging paradigm that deals with non-stationary environments during the test data adaptation. Most existing continual TTA methods are based on pseudo-labeling, but often (1) rely on overconfident pseudo-labels and (2) remain unstable under continual distribution shifts leading to error accumulation and catastrophic forgetting. To tackle these limitations, we propose *Neighbor-Filtration based Continual Test-Time Adaptation* (NF-CTTA), a reliable and memory-aware adaptation framework that addresses these challenges. NF-CTTA first calibrates pseudo-labels using class-conditional calibration error to correct over/under-confidence of the model. To further ensure reliability, we introduce an OOD Neighbor Filtration technique that selects a subset of high-confidence samples based on entropy and neighbor similarity, ensuring consistency within the semantic neighborhood. Finally, we propose a priority-guided memory buffer that retains the most informative low-entropy samples for replay, mitigating catastrophic forgetting across evolving test distributions. Extensive experiments across multiple domain shift benchmarks demonstrate that NF-CTTA achieves superior performance and stability compared to existing TTA and CTTA methods. The code is available at: https://github.com/takihasan/NF-CTTA.

CCS Concepts

• **Computing methodologies** → **Machine learning algorithms**.

Keywords

Test-Time Adaptation, Continual Learning, Pseudo-Labels.

ACM Reference Format:
Taki Hasan Rafi, Amit Agarwal, Hitesh L. Patel, and Dong-Kyu Chae. 2025. Towards Robust Continual Test-Time Adaptation via Neighbor Filtration. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3746252.3760858

*Corresponding author.

CIKM '25, Seoul, Republic of Korea

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3760858

1 Introduction

Test-Time Adaptation (TTA) aims to improve prediction performance in the unseen test data without accessing the train data during inference time. But in the real world, test data are often non-i.i.d., so there is a high discrepancy between the train (source) and test (target) data. Recent studies prove that TTA methods are robust to discrepancies that are also referred to as distribution shift [1, 11, 16, 20, 21]. Due to privacy concerns or legal constraints, it is often hard to access source training data during the model testing phase (e.g. medical domains, legal domains, recommendation systems, secure computing, etc). To tackle this limitation, TTA methods do not require source data during the test phase, which makes these methods more practical than unsupervised domain adaptation and domain generalization. Although recent studies have achieved remarkable TTA performance in multiple scenarios, most models tend to have a higher memory footprint due to the explicit backpropagation of all test samples during adaptation. Traditionally, TTA methods adapt unlabeled test samples on-the-fly. It accomplishes the source-data-free task using a pre-trained model derived from the source data. Moreover, several methods have been developed for handling distribution shifts without accessing the source data [9, 10, 18, 20, 23] at test time. On the other hand, catastrophic forgetting occurs when the model forgets previously learned knowledge while adapting to new tasks. It is more severe in TTA settings, due to encountering continuous distribution shifts during inference.

Recent continual learning-based TTA (CTTA) methods leverage pseudo-labels [2, 21]. However, naively generating pseudo-labels can lead to error accumulation and significantly reduce the performance over continual distribution shift. Although some works [2, 21, 22] have tried to retain better pseudo-labels, there is uncertainty about the models' confidence for the candidate pseudo-labels. So, models usually are not well-calibrated. Moreover, models often fail to perform well on in-distribution (ID) test samples due to the forgetting issues. However, entropy-based TTA methods [8, 20] are not always suitable to adapt to changing environments [21]. But recent works [8, 14] utilized entropy-based sample selection that achieved better results, but these samples do not preserve semantic neighborhood information so that more samples are used during adaptation, limiting computational efficiency.

Inspired by these limitations, we introduce a *neighbor-filtration based continual test-time adaptation* (NF-CTTA) framework which is robust to continuous environment shifts. We mainly focus on two challenges: (1) unreliable pseudo-labels and (2) model forgetting. To improve label quality, we explore class-conditional calibration

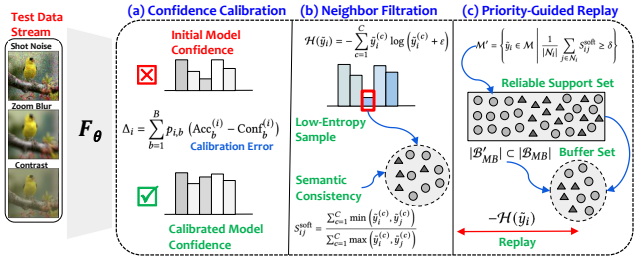

Figure 1: The Overview of our NF-CTTA framework.

to generate calibrated pseudo-labels by reflecting prediction confidence that is absent in previous methods [2, 21, 21]. We then propose an OOD neighbor filtration module that retains high-quality in-distribution samples that are semantically consistent to guide the adaptation to avoid redundancy during backpropagation. Moreover, we propose a priority-guided memory buffer to replay high-quality samples during adaptation to prevent the catastrophic forgetting problem in dynamic environments.

In summary, we have multi-fold contributions: ❶ We propose an OOD neighbor-filtration module that selects semantically consistent, low-entropy samples for adaptation and effectively reduces error accumulation in dynamically changing environments. ❷ We design a priority-guided memory buffer that retains high-confidence samples from previous test samples to avoid forgetting issues. ❸ We experimentally demonstrate that our framework can outperform SOTA methods in both accuracy and memory footprint in different dynamic TTA benchmarks.

2 Method

2.1 Task Definition

Following [21, 22], we define *continual test-time adaptation* as the process of adapting a model $f_\theta(x)$, pre-trained with parameters θ on labeled source data $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s)$, to a sequence of domains $\mathcal{D} = (\mathcal{D}_s, \mathcal{D}_2, \ldots, \mathcal{D}_T)$. Due to privacy or memory constraints, access to the source data is not available during test time. Thus, the model must make predictions on a dynamically evolving, unlabeled target domain $\mathcal{D}_T = \{x_t\}_{i=1}^{n}$ in an online streaming setting. At each time step t, the model predicts on $x_t \in \mathcal{D}_T$ and updates its parameters $\theta_t \rightarrow \theta_{t+1}$ for subsequent data from $\mathcal{D}_{t+1}$. The model is evaluated on its online predictions. Our goal is to adapt the model using only reliable online samples, rather than all samples, to ensure memory efficiency and mitigate catastrophic forgetting.

Framework Overview. We propose *neighbor-filtration based continual test time adaptation (NF-CTTA)*, a robust framework that aims to address the (1) unreliable pseudo-labels and (2) catastrophic forgetting challenges under distribution shift. Figure 1 illustrates its overview. Following similar CTTA approaches [2, 21, 22], we adopt a student-teacher scheme based on the mean teacher framework [19]. Given an input sample x_t from the online data stream originating from a random domain $\mathcal{D}_t$, the teacher model generates a pseudo-label $\hat{y}_t$ using an augmentation module, while the student model produces a pseudo-label $\bar{y}_t$, and the model is trained with consistency loss to enforce alignment between predicted labels. But labels are not always reliable due to distribution shifts; to improve label reliability, we calibrate pseudo-labels with class-conditional calibration error that ensures the model is not over- and under-confident. We perform *OOD neighbor filtration*, selecting a reliable set of test samples based on entropy and Jaccard similarity of the neighbors. This module ensures high-confidence, semantically consistent samples for adaptation. To avoid forgetting, we introduce a *prority-guided memory buffer* that retains high-priority samples based on low entropy from the past batch. These are replayed during adaptation to maintain performance on previously seen domains.

2.2 Calibrated Pseudo-Labels

As not all samples in $\mathcal{D}_t$ contribute equally to model adaptation due to varying quality, filtering becomes crucial. Existing approaches [17, 22] typically use either a fixed or dynamic confidence threshold to filter out low-quality samples, but they often overlook the calibration of confidence scores. This can bias the model toward frequent classes, incorrectly treating them as high-quality samples. Moreover, naively using soft pseudo-labels from a teacher model can cause miscalibration in the student model. To address this, we adopt class-conditional calibration error [15] to capture the extent to which the model is overconfident or underconfident.

$$\Delta_i = \sum_{b=1}^{B} p_{i,b} \left(\mathrm{Acc}_b^{(i)} - \mathrm{Conf}_b^{(i)} \right), \qquad p_{i,b} \equiv \frac{N_{i,b}}{N_i}. \tag{1}$$

Here, the confidence interval [0,1] is split into bins, which is represented by B (we set B=10). $p_{i,b}$ is a weight of true class-i samples that fall into the confidence bin b, and $N_{i,b}$ is the number of samples whose predicted confidence falls under the bin b. N_i is the total number of samples in class-i. The optimal Δ_i score should be 0 to expect the pseudo-label predictions are well-calibrated.

Unlike [12], where $|\mathrm{Acc}_b^{(i)} - \mathrm{Conf}_b^{(i)}|$ gives only the magnitude information of the calibration error, our method can determine whether the model is overconfident or underconfident with Δ_i by:

$$\mathrm{Bias}_{i,b} = \begin{cases} \text{over-confident} & \text{if } \mathrm{Conf}_b^{(i)} > \mathrm{Acc}_b^{(i)}, \\ \text{under-confident} & \text{otherwise.} \end{cases}$$

We use the calibration error to adjust soft-pseudo labels by the teacher model, $\hat{y}_t$. Thus, we obtain calibrated pseudo-labels $\tilde{y}_t$ as:

$$\tilde{y}_t = \hat{y}_t + \gamma \Delta. \tag{2}$$

where, Δ_i is a vector that concatenates per-class calibration error values, and γ is a controlling factor for calibration error.

2.3 OOD Neighbor Filtration

To improve pseudo-label reliability, we propose *OOD Neighbor Filtration*, which selects a subset of samples $\mathcal{B}_s$ from the test-time batch $\mathcal{B}$ for adaptation. This subset contains the most informative and reliable candidates, reducing the influence of noisy or OOD samples. We define a *support neighbor set* $\mathcal{M} \subset \mathcal{B}$ by filtering out high-entropy (i.e., uncertain or OOD) samples using the normalized entropy [20] of the calibrated soft pseudo-labels $\{\tilde{y}_i \in \mathbb{R}^C\}_{i=1}^{N}$:

$$\mathcal{H}(\tilde{y}_i) = -\sum_{c=1}^{C} \tilde{y}_i^{(c)} \log \left(\tilde{y}_i^{(c)} + \varepsilon \right), \tag{3}$$

where C is the number of classes and ϵ is used for numerical stability. This entropy score measures the uncertainty of the model, so high-entropy samples are considered as OOD samples. Then we define a support neighbor set $\mathcal{M}$ that filters the high-entropy samples below a threshold τ:

$$\mathcal{M} = \{\tilde{y}_i \mid \mathcal{H}(\tilde{y}_i) \leq \tau\}. \tag{4}$$

To further refine $\mathcal{M}$, we compute pairwise *soft Jaccard similarity* between samples to quantify semantic consistency:

$$S_{ij}^{\text{soft}} = \frac{\sum_{c=1}^{C} \min\left(\tilde{y}_i^{(c)}, \tilde{y}_j^{(c)}\right)}{\sum_{c=1}^{C} \max\left(\tilde{y}_i^{(c)}, \tilde{y}_j^{(c)}\right)}. \tag{5}$$

where S_{ij}^{soft} indicates the semantic similarity of the predicted class distribution of two samples; higher S_{ij}^{soft} indicates overlap of the predicted class confidence, which determines the better reliability of the pseudo-labels. Intuitively, if a sample is surrounded by more similar samples, the sample is more likely to be reliable. Based on this criterion, we construct a more reliable support neighbor set $\mathcal{M}'$, which can be defined as:

$$\mathcal{M}' = \left\{ \tilde{y}_i \in \mathcal{M} \;\middle|\; \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} S_{ij}^{\text{soft}} \geq \delta \right\}, \tag{6}$$

where $\mathcal{N}_i \subset \mathcal{M}$ denotes the neighbors of sample i and δ is a similarity threshold. We then use $\mathcal{M}'$ to compute a consistency loss between the fixed teacher and the adapting student model:

$$\mathcal{L}_{\text{cons}} = \frac{1}{|\mathcal{M}'|} \sum_{\tilde{y}_i \in \mathcal{M}'} \text{KL}\left(\tilde{y}_i^{\text{teacher}} \,\middle\|\, \tilde{y}_i^{\text{student}}\right). \tag{7}$$

This loss ensures that only confident, in-distribution samples guide the student model, enhancing stable adaptation while keeping the teacher model fixed.

2.4 Priority-Guided Memory Buffer

In TTA settings, the performance can significantly drop due to catastrophic forgetting of the adapted test samples [2, 14]. To overcome this limitation, we introduce a *priority-guided memory buffer* to avoid catastrophic forgetting of the model during adaptation. Unlike adopting the current batch, our buffer holds the informative samples that can help the model revisit past knowledge without forgetting. This helps the model avoid overfitting and gradually adapt to the distribution shifts. We construct a reliable support neighbor set $\mathcal{M}'$; from this set, we select a few samples that are most representative of each class in a fixed-memory buffer based on a priority-based selection. Each of the samples in $\mathcal{M}'$ is assigned a priority score based on its entropy:

$$\text{Priority Score}(\tilde{y}_i) = -\mathcal{H}(\tilde{y}_i). \tag{8}$$

Higher priority is given to the samples with lower entropy scores, and these samples are retained in the buffer $\mathcal{B}_{MB}$. When the samples exceed the fixed memory size, we exclude samples with the higher entropy among the samples in the buffer. This process ensures the most informative samples are preserved for reply.

During the model updates, we sample a subset $\mathcal{B}'_{MB}$ from the buffer, $|\mathcal{B}'_{MB}| \subset |\mathcal{B}_{MB}|$ and compute a KL-divergence based memory buffer loss $\mathcal{L}_{MB}$:

$$\mathcal{L}_{\text{MB}} = \frac{1}{|\mathcal{B}_{\text{MB}'}|} \sum_{\tilde{y}i \in \mathcal{B}_{\text{MB}'}} \text{KL}\left(\tilde{y}_i^{\text{teacher}} | \tilde{y}_i^{\text{student}}\right). \tag{9}$$

$\mathcal{L}_{MB}$ contributes as a regularizer to mitigate the overfitting to the incoming batch; it helps to maintain consistent performance in in-distribution (ID) samples that are already seen by the model. It also balances the trade-off between plasticity and stability by replaying high-priority samples which ensure dynamic model adaptation.

Training and Inference. We use a combination of consistency loss for both reliable pseudo-label samples and memory buffer loss for the priority-guided memory buffer module. So, the total loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cons}} + \lambda \mathcal{L}_{\text{MB}} \tag{10}$$

where λ balances both terms.

3 Experiments

3.1 Experimental Setup

Dataset. We evaluate our framework with three widely used benchmarks: CIFAR-10-C, CIFAR-100-C, and Tiny-ImageNet-C [4]. (CIFAR-10/100 [6] and Tiny-ImageNet [7] are used for training). Due to GPU resource limitations, we cannot evaluate on the original ImageNet-C. All three datasets contain 15 different corruptions and each at 5 different levels of severity. We organize our experiments based on these questions: (1) How does NF-CTTA perform in comparison with other CTTA methods? (2) Can NF-CTTA learn dynamic environments without forgetting?

Implementation Details. For CIFAR 10/100-C, we use ResNet-26 and ResNet-34 [3] for Tiny-ImageNet-C. We use a batch size of 32 for CIFAR-10/100-C and 64 for Tiny-ImageNet-C. For test-time adaptation, we utilize the SGD optimizer with Nesterov momentum [5]. During both pre-training and adaptation, the learning rate is set to 0.001 with a linear decay and the epoch is 75. δ is set to 0.7 for CIFAR-10/100-C and 0.6 for Tiny-ImageNet-C due to higher class diversity. To make a fair comparison with the baselines, we employ the same identical pre-trained models and hyperparameters. We follow the baseline settings from [13], and evaluate all the comparisons in average online prediction error rate $\mathcal{E}(\psi)$.

3.2 Results

Instantaneously Changing Setup. In Table 1, we show the results for CIFAR-10/100-C and Tiny-ImageNet-C datasets under the highest corruption severity (level 5) in the standard domain sequence. Our method exhibits an average online error of *17.23%*, *39.27%*, and *62.56%* respectively, which demonstrates a significant outperformance with respect to other baseline methods. Compared with similar methods like EATA, there is a 5.59% and 17.03% improvement on CIFAR-10/100-C. Moreover, in the most challenging Tiny-ImageNet-C, we also show better the average $\mathcal{E}(\psi)$ rate. It verifies the effectiveness of our calibrated pseudo-labels and semantically consistent samples.

In-Distribution (ID) Forgetting. We investigate its anti-forgetting capacity by comparing with EATA. Following the setup from [14], we evaluate lifelong (continual) adaptation without resetting model

Table 1: Comparative results in terms of online error $\mathcal{E}(\psi)$ (%) on *CIFAR-10/100-C* and *Tiny-ImageNet-C* with the highest corruption severity on the instantaneously changing setup.

Dataset	Method	gauss	shot	impul	defoc	glass	motn	zoom	snow	frost	fog	brit	contr	elast	pixel	jpeg	$\mathcal{E}(\psi)\downarrow$
CIFAR-10-C	ERM	73.06	67.73	73.38	23.71	64.87	34.53	27.08	33.41	47.54	19.49	10.97	23.14	39.14	73.25	36.70	43.20
	EATA[14] (*ICML'22*)	23.21	26.45	28.13	18.42	38.18	24.07	22.90	26.75	27.76	21.69	19.83	19.89	30.67	26.10	25.38	23.82
	TENT[20] (*ICLR'21*)	27.83	23.08	29.89	20.60	42.48	28.79	21.66	31.42	39.48	29.88	31.71	41.08	49.51	50.70	50.34	35.67
	CoTTA[21] (*CVPR'22*)	25.63	22.46	27.09	19.75	35.35	31.33	25.16	34.24	38.85	29.15	31.85	33.40	33.22	29.04	28.87	26.51
	DSS[22] (*WACV'24*)	24.21	22.46	26.78	18.67	35.37	31.33	24.82	33.10	39.35	30.28	28.78	33.40	31.28	27.94	26.78	25.92
	CasTTA[13] (*CIKM'24*)	28.64	25.36	32.38	15.52	40.55	19.55	16.77	20.65	20.78	19.81	11.34	16.86	28.36	24.16	24.70	22.99
	NF-CTTA (Ours)	**21.04**	**20.64**	**25.67**	**13.23**	**29.05**	**13.70**	**14.86**	**17.89**	**18.45**	**15.23**	**10.56**	**12.45**	**21.39**	**18.26**	**20.41**	**17.23**
CIFAR-100-C	ERM	94.24	91.36	91.52	55.00	87.42	62.58	56.06	67.12	76.52	57.58	42.82	66.97	65.15	90.91	67.73	71.51
	EATA[14] (*ICML'22*)	64.23	67.28	67.90	56.87	73.78	60.25	57.28	55.67	50.00	60.97	39.28	70.01	60.23	55.10	53.28	56.30
	TENT[20] (*ICLR'21*)	76.52	71.21	68.18	48.79	71.52	52.88	45.15	55.61	53.33	52.42	41.21	54.55	55.00	52.27	59.39	57.20
	CoTTA[21] (*CVPR'22*)	68.79	70.61	69.39	57.88	70.00	61.36	55.61	61.36	56.67	62.27	45.00	75.61	64.09	60.00	60.30	62.60
	DSS[22] (*WACV'24*)	65.64	69.09	69.39	57.88	71.45	59.20	54.25	61.36	52.34	61.90	41.29	72.56	62.29	60.00	56.23	59.70
	CasTTA[13] (*CIKM'24*)	66.42	64.78	65.65	44.33	67.20	27.89	46.01	54.11	35.48	53.75	39.85	49.87	54.66	60.07	55.87	55.11
	NF-CTTA (Ours)	**43.31**	**44.92**	**47.23**	**31.00**	**49.87**	**21.71**	**29.87**	**32.81**	**21.78**	**38.12**	**23.78**	**30.19**	**41.87**	**41.89**	**38.28**	**39.27**
Tiny ImageNet-C	ERM	86.06	83.94	88.60	92.27	77.39	77.27	65.41	61.21	57.42	75.45	52.72	95.45	81.27	64.55	48.94	76.49
	EATA[14] (*ICML'22*)	80.10	77.24	84.56	80.19	77.89	71.29	70.78	72.30	67.89	65.78	51.72	90.42	62.24	77.23	73.80	68.81
	TENT[20] (*ICLR'21*)	85.22	82.79	88.23	84.40	91.61	82.36	83.49	84.17	86.33	83.14	96.48	96.48	89.76	88.28	89.90	86.42
	CoTTA[21] (*CVPR'22*)	83.49	80.37	87.86	79.22	89.18	69.65	71.10	70.83	69.49	74.49	64.67	93.07	73.35	65.33	68.00	76.01
	DSS[22] (*WACV'24*)	81.32	82.24	83.79	85.82	79.73	73.03	78.03	70.83	58.03	74.24	41.06	91.81	70.18	66.36	63.03	69.34
	CasTTA[13] (*CIKM'24*)	74.62	70.12	77.80	77.56	83.63	54.13	54.04	59.22	53.69	63.77	49.34	91.72	63.29	53.84	49.56	65.28
	NF-CTTA (Ours)	**71.52**	**65.78**	**73.28**	**72.90**	**75.90**	**49.81**	**49.20**	**51.66**	**49.20**	**61.29**	**48.14**	**87.23**	**59.80**	**49.28**	**45.12**	**62.56**

Table 2: Results on *Tiny-ImageNet-C* (averaging severity level-5) for the average clean accuracy and average corrupted accuracy (%). Results are reported after OOD adaptation. Clean accuracy (before) means without OOD adaptation.

Method	Clean Acc. (Before)	Clean Acc.	Corr Acc.
TENT	72.45	68.25	23.55
EATA	77.54	76.14	31.56
NF-CTTA	**81.26**	**81.00**	**39.24**

Table 3: Results on *CIFAR-10-C* for the average adaptation time per corruption and average memory usage in MB.

Method	Adaptation Time (s) ↓	Memory (MB) ↓
TENT	18	986.23
EATA	17	582.87
CoTTA	24	1735.34
DSS	19	879.87
CasTTA	**7**	406.60
NF-CTTA	9	**382.56**

parameters. As shown in Table 2, our method achieves better results than EATA on similar settings (ResNet-34 backbone), while maintaining ID clean accuracy due to our priority-guided replay of the in-distribution (ID) low-entropy samples. Similar performance in both before/after adaptation in ID-clean accuracy demonstrates our model's stability and effectiveness in retaining past knowledge.

Adaptation Time & Memory. Table 3 reports adaptation time (middle) and memory (right). Our method significantly reduces the adaptation time and memory, due to partial sample selection for adaptation unlike CoTTA and fewer samples than EATA. However, adaptation time remains the second best after CasTTA due to our priority-guided buffer that eventually overcomes the forgetting.

Ablation on Different Components. CoTTA naively utilizes noisy pseudo-labels for distribution matching. Our pseudo-label calibration module single-handedly improves performance on all benchmarks (see Table 4) over CoTTA, which implies the effect of model confidence to handle noisy pseudo-labels. Furthermore, combining the other modules (CPL + NF + PB) further improves performance, demonstrating the effectiveness of each proposed module.

Table 4: Online prediction error $\mathcal{E}(\psi)$ on corrupted datasets. Here, we denote CPL = Calibrated Pseudo-Labels, NF = Neighbor Filtration, PB = Priority Buffer.

$\mathcal{E}(\psi)\downarrow$	CIFAR-10-C	CIFAR-100-C	Tiny-Imag.-C
CoTTA	26.51	62.60	76.01
NF-CTTA (w/ CPL)	24.75	57.28	68.71
NF-CTTA (w/ CPL + NF)	21.76	46.87	65.67
NF-CTTA (w/ CPL + NF + PB)	**17.23**	**39.27**	**62.56**

4 Concluding Remarks

In this paper, we propose a novel continual test-time adaptation method, *NF-CTTA*, which employs neighbor filtration to obtain low-entropy, high-quality samples that preserve semantic consistency for robust adaptation. To prevent forgetting in continual environments, a priority-guided buffer selectively replays samples during adaptation. Experimental results demonstrate that NF-CTTA outperforms state-of-the-art baselines on corrupted benchmark settings. Inherently, TTA methods are often computationally heavy, and our neighbor similarity imposes additional computational overhead. Future works can focus on reducing further time complexity in TTA methods.

Acknowledgement

This work was supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(RS-2025-25422680, Metacognitive AGI Framework and its Applications, and RS-2020-II201373, Artificial Intelligence Graduate School Program (Hanyang University)).

GenAI Usage Disclosure

The authors declare that they used GenAI tools only for *minor* language editing, content refinement, and formatting. The authors take full responsibility for the contents of this paper.

References

[1] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. 2022. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8344–8353.

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

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

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

[5] Nikhil Ketkar. 2017. Stochastic gradient descent. In *Deep learning with Python: A hands-on introduction*. Springer, 113–132.

[6] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[7] Yann Le and Xuan Yang. 2015. Tiny imagenet visual recognition challenge. *CS 231N* 7, 7 (2015), 3.

[8] Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. 2024. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. *arXiv preprint arXiv:2403.07366* (2024).

[9] Jian Liang, Dapeng Hu, and Jiashi Feng. 2020. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*. PMLR, 6028–6039.

[10] Chaithanya Kumar Mummadi, Robin Hutmacher, Kilian Rambach, Evgeny Levinkov, Thomas Brox, and Jan Hendrik Metzen. 2021. Test-time adaptation to distribution shift by confidence maximization and input transformation. *arXiv preprint arXiv:2106.14999* (2021).

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

[12] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 29.

[13] Kien X Nguyen, Fengchun Qiao, and Xi Peng. 2024. Adaptive Cascading Network for Continual Test-Time Adaptation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 1763–1773.

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

[15] François Porcher, Camille Couprie, Marc Szafraniec, and Jakob Verbeek. 2024. Better (pseudo-) labels for semi-supervised instance segmentation. *arXiv preprint arXiv:2403.11675* (2024).

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

[17] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems* 33 (2020), 596–608.

[18] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*. PMLR, 9229–9248.

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

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

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

[22] Yanshuo Wang, Jie Hong, Ali Cheraghian, Shafin Rahman, David Ahmedt-Aristizabal, Lars Petersson, and Mehrtash Harandi. 2024. Continual test-time domain adaptation via dynamic sample selection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1701–1710.

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