Maintaining Consistent Inter-Class Topology in Continual Test-Time Adaptation

Supplementary Material

1. Random Domain shift CTTA

Considering the randomness of domain shifts, we conducted experiments on CIFAR10-C, CIFAR100-C, and ImageNet-C with multiple sequences of various corruption types at severity level 5. Specifically, We randomly generate 10 domain shift sequences. In each sequence, we do not modify any hyperparameters. We report the mean error rate and variance of various methods on the CTTA task across 10 sequences. As is shown in the Tab 1, TCA reduces the average error rate to 9.2%, 8.9%, and 5.1% on the random order CIFAR10-to-CIFAR10C, CIFAR100-to-CIFAR100C, and ImageNet-to-ImageNetC benchmarks, respectively. This implies that the order of domain changes does not affect the TCA's ability to maintain a stable inter-class topological structure.

Avg. Error (%)	Source	TENT	CoTTA	RMT	TCA
CIFAR10-C	43.5	20.1	16.3	15.6	**14.8** ± 0.15
CIFAR100-C	46.4	61.3	32.6	30.2	**29.7** ± 0.11
ImageNet-C	83.0	62.8	62.6	60.1	**59.4** ± 0.23

Table 1. Average Error (%) on CIFAR10-C, CIFAR100-C, and ImageNet-C on random TTA setup.

2. Gradual Test Time Adaptation

Following CoTTA [7], we show gradual corruption results instead of constant severity in the major comparison. Specifically, each corruption adapts the gradual changing sequence:1 → 2 → 3 → 4 → 5 → 4 → 3 → 2 → 1, where the severity level is the lowest 1 when the corruption type changes, therefore, the type is also gradual. As shown in Tab 2, TCA achieves superior performance in this setup, reducing the average error rate to 21.1%, 3.1%, and 2.9% on the gradual CIFAR10-to-CIFAR10C, CIFAR100-to-CIFAR100C, and ImageNet-to-ImageNetC benchmarks, respectively. This indicates that, despite variations in the degree of domain degradation, the TCA does not become trapped in local optima due to less degraded domains. Instead, TCA achieves more accurate centroids, providing a more stable inter-class topological structure during testing, resulting in superior performance.

Avg. Error (%)	Source	TENT	CoTTA	RMT	TCA
CIFAR10C	24.7	20.4	10.9	9.3	**8.6**
CIFAR100C	33.6	74.8	26.3	26.4	**25.8**
ImageNet-C	58.4	46.4	38.8	39.3	**38.1**

Table 2. Classification error rate (%) for the gradual CIFAR10-to-CIFAR10C, CIFAR100-to-CIFAR100C, and ImageNet-to-ImageNet-C benchmark averaged over all 15 corruptions. The severity level changes gradually between the lowest and the highest.

3. Experiments on Segmentation CTTA

Cityscapes-to-ACDC. This dataset is designed for continuous semantic segmentation tasks [7] in autonomous driving. The Cityscapes dataset serves as the source domain, providing pre-trained segmentation models, while the Adverse Conditions dataset (ACDC) [5] includes images collected under four different weather conditions: fog, night, rain, and snow, representing the dynamic target domain. For each condition, 400 unlabeled images are used for adaptation. Notably, the ACDC and Cityscapes datasets share identical semantic classes for evaluation. To simulate the continuous distribution changes encountered in real-world scenarios, we repeat the same target domain sequence ten times (*e.g.* a total of 40 transitions: Fog → Night → Rain → Snow → Fog → ...). This setup offers a long-term perspective for evaluating the performance of various adaptation methods.

We report results based on the mean intersection over union (mIoU) metric for the complex continual test-time semantic segmentation Cityscapes-to-ACDC task. As shown in Tab. 3, Tent exhibits a performance decline in long sequence tasks, highlighting the common issue of model degradation over time. CoTTA and BeCoTTA offer a more stable CTTA process. However, they do not improve model performance with each adaptation cycle. SVDP also demonstrates instability across multiple adaptation rounds. In contrast, TCA maintains the latent inter-class topological structure during testing, achieving a stable testing process and a 5.1% relative improvement in mIoU compared to the baseline CoTTA, averaged across ten rounds.

4. Inter-class Feartue Uniformity Analysis

To further confirm the uniformity of inter-class feature distribution, we select features from different classes distributed on the hypersphere. Following the setup in section 4.4, we randomly choose 10 batches from the beginning, middle, and end domains of CTTA. We visualize features of classes 3, 6, and 9 from CIFAR10-C. As shown in Fig. 1, in both CoTTA and TCA methods, the intra-class feature distribution is overly dispersed, with features scattered across various regions of the sphere, which easily leads to outlier

Time	t →																
Round	1				4				7				10				All
Condition	Fog	Night	rain	snow	Fog	Night	rain	snow	Fog	Night	rain	snow	Fog	Night	rain	snow	Mean
Source	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	56.7
BN Stats Adapt	62.3	38.0	54.6	53.0	62.3	38.0	54.6	53.0	62.3	38.0	54.6	53.0	62.3	38.0	54.6	53.0	52.0
TENT-continual [6]	69.0	40.2	60.1	57.3	66.5	36.3	58.7	54.0	64.2	32.8	55.3	50.9	61.8	29.8	51.9	47.8	52.3
CoTTA [7]	70.9	41.2	62.4	59.7	70.9	41.0	62.7	59.7	70.9	41.0	62.8	59.7	70.8	41.0	62.8	59.7	58.6
BEcoTTA [3]	72.0	**45.4**	63.7	60.0	71.7	**45.4**	63.6	60.1	71.8	**45.4**	63.7	60.1	71.7	**45.3**	63.6	60.0	60.2
VDP [2]	70.5	41.1	62.1	59.5	70.4	41.1	62.2	59.4	70.4	41.0	62.6	59.4	70.4	41.1	62.5	59.4	58.2
TCA	**72.2**	44.5	**65.4**	**63.0**	**72.6**	44.7	**66.1**	63.2	**72.4**	44.4	**65.9**	63.3	**72.5**	44.4	**65.7**	**63.8**	**61.6**

Table 3. Semantic segmentation results (mIoU in %) on the Cityscapes-to-ACDC online continual test-time adaptation task. We continually evaluate the four test conditions ten times to evaluate the long-term adaptation performance. For brevity, we only show the continual adaptation results in the first, fourth, seventh, and last round. All results are evaluated based on the Segformer-B5 architecture. Bold text indicates the best performance.

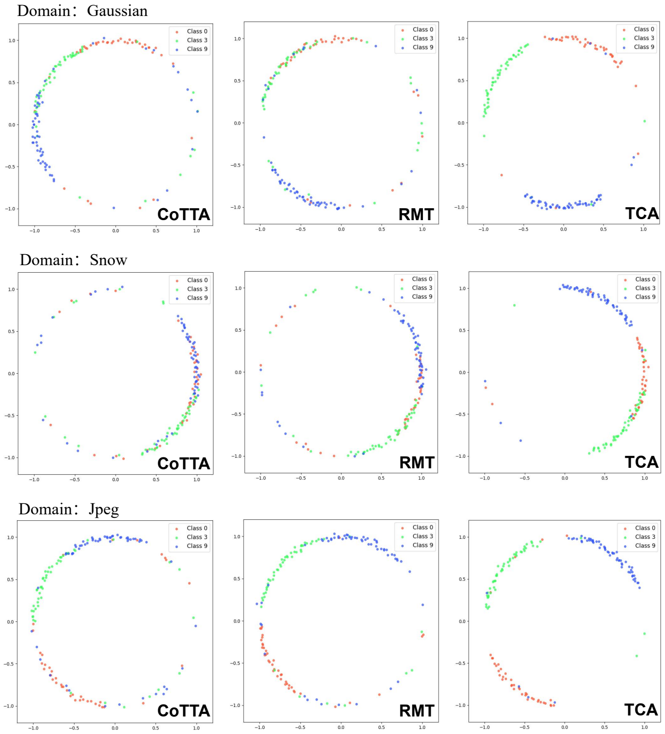

Figure 1. Feature distribution of classes 0, 3, and 9 in $\mathbb{R}^2$ using Gaussian kernel density estimation (KDE) in the domain Gaussian, Snow and Jpeg (which means the beginning, middle, and end of the CTTA process).

noise. Additionally, excessive overlap of inter-class features interferes with the decision boundary, disrupting the inter-class topological structure. In contrast, TCA achieves a stable and uniform feature distribution, resulting in compact intra-class features while maintaining a uniform inter-class distribution, thus preserving a stable inter-class topological structure.

5. Inter-class Topology Preservation

To intuitively observe how TCA effectively maintains inter-class topological structures, we visualize three methods on the CIFAR10-CIFAR10C CTTA task. Following the setup in Section 4.5, we randomly select features from one batch of data in the Gaussian, Snow, and Jpeg domains for t-SNE visualization. We display only the distribution of centroids, omitting fully connected edges for clarity. Connections are drawn between centroids closest in the source domain to observe changes in topological structure. As shown in Fig. 2, CoTTA and RMT struggle to maintain stable inter-class topological structures during adaptation. Edges between centroids intersect in the two-dimensional space, indicating a significant overlap of surrounding features, which interferes with the model's decision-making. In contrast, TCA consistently maintains a relatively stable topological structure. Although the lengths of edges vary, their relative magnitudes remain as consistent as possible.

Method	**Class Decremental**	**Class Incremental**
CoTTA [7]	91.62	68.16
RMT [1]	93.31	78.31
TCA	**95.28**	**86.89**

Table 4. The classification (accuracy in %) validation experiments of TCA in class incremental and decremental scenarios.

6. Setting with Varying Class Numbers

In this section, we consider validation experiments for TCA in the scenario of varying the number of classes. In class-decremental settings, TCA only needs to maintain topological relationships among the remaining classes. In class-incremental settings, TCA easily adds new graph nodes to the topology, with new nodes corresponding to the prototypes of the new classes. To illustrate this, we conduct experiments on MNIST as shown in Tab. 4, implementing 9 +

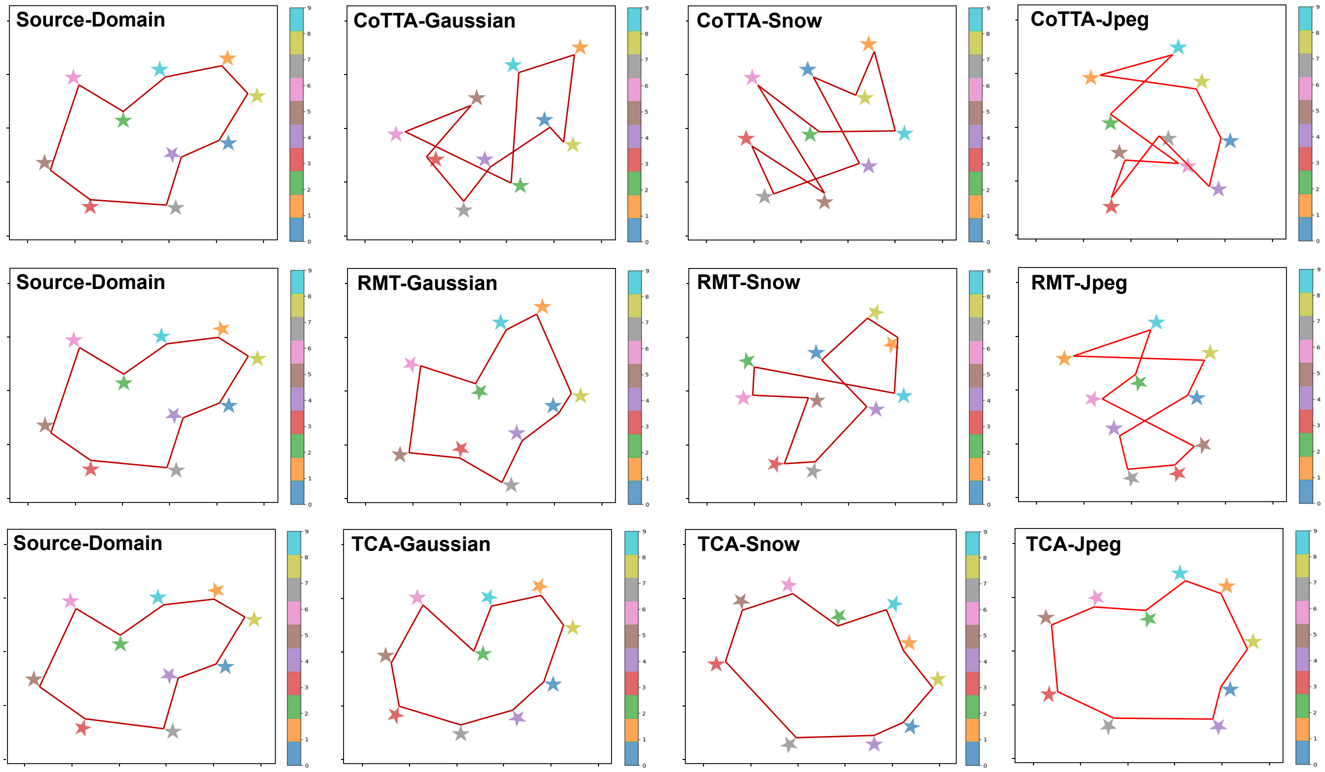

Figure 2. The visualization of centroid distribution in the CTTA process using three methods. To enhance observation of the topological structure, only partial edges are displayed, omitting fully connected edges.

1 for incremental and 10 - 1 for decremental scenarios. Note that we use a nearest-neighbor classifier to avoid confusion with the classifier.

7. Compare with TTA Methods

In this section, we apply TCA in the TTA setting and conduct analogous experiments following the TTAB [4] protocol. Tab. 5 represents TTA for common distribution shifts, which consists of two parts: synthetic covariate shift (CIFAR10-C) and domain generalization (OfficeHome and PACS). Tab. 6 represents spurious correlation shifts, where certain features are spuriously correlated with the target variable in the training data but not in the test data. The experimental results demonstrate that TCA remains effective even in the resetting TTA scenario.

Method	CIFAR10-C	OfficeHome	PACS
TTT [4]	20.9	40.2	25.3
CoTTA [7]	25.3	53.7	28.6
TCA	**11.7**	**34.7**	**19.4**

Table 5. The classification (error rate in %) validation experiments of TCA with TTA methods in common distribution shifts and domain generalization scenarios.

Method	ColoredMNIST	Waterbirds
TTT [4]	78.1	28.2
CoTTA [7]	72.6	31.7
TCA	**58.2**	**26.2**

Table 6. The classification (error rate in %) validation experiments of TCA with TTA methods in spurious correlation shifts, scenarios.

8. Hyperparameter Sensitivity

We consider four hyperparameters: α, β, λ_1, and λ_2. First, considering that α in Equation 3 is designed to maintain the graph topology as non-empty, leading to two scenarios for its value. If a class in the current batch is non-empty, we set α in the update equation for that class to 0.001. As shown in Tab 7, we provide the ablation of α on CIFAR10-C. If the class in the current batch is empty, we supplement graph nodes with prior nodes, setting α to 1.

Then, we conduct an ablation study on β in Equation 9, selecting its value from $[0, 1.0]$ at intervals of 0.1. We provide the ablation of β on CIFAR10-C. As shown in Tab 8, the best performance is achieved when $\beta = 0.2$. Notably, the model performs poorly when $\beta = 0$ and $\beta = 1.0$, in-

Value of α	0.01	0.015	**0.001**	0.0005	0.00001
Error (%)	14.9	14.8	**14.7**	14.8	14.8

Table 7. Choice of α in Equation 3.

dicating that integrating both prior and current distributions better helps the model maintain a stable inter-class topology.

Value of β	0.0	0.1	**0.2**	0.3	0.4	0.5
Error (%)	15.5	15.0	**14.7**	14.9	15.0	14.9
Value of β	0.6	0.7	0.8	0.9	1.0	-
Error (%)	15.2	15.3	15.4	15.4	15.8	-

Table 8. Choice of β in Equation 9.

Finally, we conduct a detailed hyperparameter search for λ_1 and λ_2. Referring to the suggestions in UA, which utilize $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ for fine-tuning, we employ smaller hyperparameter coefficients. As shown in Tab 9, we provide 25 linear searches over λ_1 and λ_2.

$\lambda_2 \backslash \lambda_1$	0.10	0.125	**0.15**	0.175	0.20
0.01	15.2	15.1	15.2	15.3	15.3
0.015	15.1	15.0	15.0	15.2	15.3
0.02	15.0	15.0	14.8	15.1	15.0
0.025	15.0	14.9	**14.7**	14.9	15.0
0.03	14.9	14.9	14.8	14.9	14.9

Table 9. Linear combinations of λ_1 and λ_2.

References

[1] Mario Döbler, Robert A Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. In *CVPR*, pages 7704–7714, 2023. 2

[2] Yulu Gan, Yan Bai, Yihang Lou, Xianzheng Ma, Renrui Zhang, Nian Shi, and Lin Luo. Decorate the newcomers: Visual domain prompt for continual test time adaptation. In *AAAI*, pages 7595–7603, 2023. 2

[3] Daeun Lee, Jaehong Yoon, and Sung Ju Hwang. Becotta: Input-dependent online blending of experts for continual test-time adaptation. *arXiv preprint arXiv:2402.08712*, 2024. 2

[4] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34: 21808–21820, 2021. 3

[5] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10765–10775, 2021. 1

[6] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020. 2

[7] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, pages 7201–7211, 2022. 1, 2, 3