CLASS TOPOLOGY SURVIVES CORRUPTION: PRIOR, MONITOR, AND REPAIR FOR 3D CONTINUAL TEST-TIME ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual test-time adaptation (CTTA) aims to adapt a source-trained model to continuously evolving target distributions without access to source data, target labels, or domain-boundary information. Most existing CTTA methods rely on computationally expensive mean-teacher frameworks. Moreover, they often require explicit distribution-shift detection and treat each target distribution independently, limiting their ability to exploit information shared across domains. In this work, we show that, despite substantial accuracy degradation, the relational topology among classes remains remarkably stable across diverse 3D corruptions. Motivated by this observation, we propose **To**pology-Anchored **Co**ntinual **T**est-**T**ime **A**daptation (ToCoTTA), which leverages this class topology as a stable structural anchor throughout continual adaptation. Specifically, ToCoTTA derives a reference class topology from the source classifier and uses it in three complementary ways: as a *prior* to refine noisy online class prototypes, as a *monitor* to quantify the reliability of ongoing adaptation, and as a *repair* signal to regulate updates and recover from accumulated parameter drift. By providing a unified structural signal for both prediction and adaptation, class topology enables ToCoTTA to remain robust under continually evolving target distributions. Extensive experiments on multiple 3D point cloud benchmarks demonstrate consistent improvements over existing CTTA methods with negligible computational overhead.

1 INTRODUCTION

Deep neural networks achieve strong performance across various 3D point cloud tasks (Qi et al., 2017; Wang et al., 2019; Pang et al., 2022), but typically assume that target (test) data follow the same distribution as the source (training) data. In real-world deployments, however, this assumption often breaks down due to sensor- and environment-induced distribution shifts, such as sparsity and occlusion, resulting in substantial performance degradation. Given the diversity of potential distribution shifts, it is impractical to account for all of them during training.

Test-time adaptation (TTA) addresses unforeseen distribution shifts by adapting a source-trained model online using only unlabeled target samples at inference time. Existing TTA methods (Wang et al., 2021; Maharana et al., 2025; Kim et al., 2025; Bahri et al., 2025) typically rely on self-supervised objectives such as pseudo-labeling (Lee et al., 2013) or entropy minimization. They often update only a small subset of parameters, such as the affine parameters of normalization layers or learnable prompts (Shu et al., 2022; Xiao et al., 2025; Ahamed et al., 2026).

However, conventional TTA is often evaluated in an episodic setting with a single fixed target distribution. Such a setting rarely reflects real-world deployment, where target distributions evolve continuously and the model must adapt without knowing *when* the current domain ends and a new one begins. Continual test-time adaptation (CTTA) (Wang et al., 2022; Niu et al., 2022; Gong et al., 2022; Song et al., 2023; Lee et al., 2024; Ni et al., 2025) considers this more realistic setting by continually adapting the model across a sequence of target domains, without resets, source data, target labels, or domain-boundary information. This continual adaptation poses two key challenges: error accumulation from unreliable self-supervision over time, and catastrophic forgetting caused by repeated updates that overwrite previously acquired knowledge.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

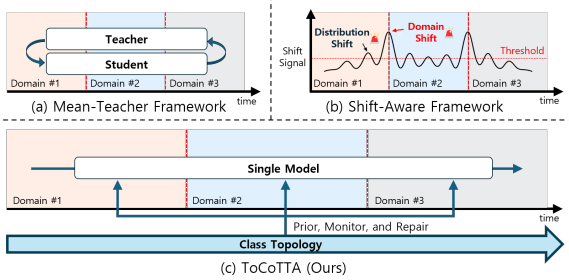


Figure 1: Existing CTTA methods commonly rely on (a) teacher–student frameworks or (b) shift-aware frameworks that explicitly detect distribution shifts. In contrast, ToCoTTA employs a single model and leverages class topology as a stable structural anchor throughout continually evolving target distributions.

Most CTTA methods rely on mean-teacher frameworks (Tarvainen & Valpola, 2017), often coupled with model restoration or regularization. Building on this paradigm, distribution-aware methods (Yang et al., 2024; Liu et al., 2024a;b) leverage information about distribution shifts to guide adaptation. More recent approaches (Yoo et al., 2024; Zhao et al., 2026; Park et al., 2025) go further by explicitly detecting *when* the domain changes and adjusting the adaptation process accordingly. Despite their effectiveness, maintaining separate teacher and student networks incurs substantial memory and computational costs, reducing inference efficiency. Domain shift detection can further increase this overhead, as distribution shifts are often quantified using uncertainty estimates obtained from multiple stochastic forward passes with MC-Dropout (Gal & Ghahramani, 2016). Such detection also introduces an additional threshold hyperparameter that may require careful tuning. More importantly, partitioning the test stream according to detected domain shifts may overlook knowledge that is shared across domains. Even when using domain-shared knowledge, existing methods often capture it implicitly through learned components, such as adapters or experts (Liu et al., 2024b; Zhao et al., 2026), without explicitly characterizing what remains invariant across domains.

To address these limitations, we propose **To**pology-anchored **Co**ntinual **T**est-**T**ime **A**daptation (ToCoTTA), which explicitly identifies and leverages knowledge shared across domains. Our key observation is that the relational structure among classes, which we term *class topology*, remains highly stable across corruption types and severities. We define class topology through cosine affinities between class prototypes in the feature space. Across diverse corruptions, class topology remains highly correlated with the source topology, with the correlation decreasing substantially more slowly than classification accuracy as corruption severity increases. As illustrated in Fig. 1, ToCoTTA maintains a single model and continuously leverages class topology across changing domains, without relying on teacher–student models or detecting distribution shifts.

Motivated by this observation, ToCoTTA leverages class topology in three complementary roles. As a **Prior**, it maintains running class prototypes over the test stream and refines them toward the source affinity structure before prediction. The refined prototypes define an auxiliary prototype head, whose predictions are fused with those of the original classification head for final predictions. As a **Monitor**, it measures the alignment between the current prototype affinity and the source affinity. The resulting correlation serves as a reliability signal that decreases when prototype estimates are corrupted by erroneous adaptation. As a **Repair**, this reliability signal modulates the adaptation objective to suppress further error accumulation. Moreover, it controls soft restoration of the trainable normalization parameters toward their source values, thereby reducing accumulated adaptation errors. By preventing excessive drift from the source initialization, this restoration also helps mitigate catastrophic forgetting. Notably, the same topology information supports prototype refinement, reliability estimation, and model restoration.

The main contributions can be summarized as follows:

- We identify class topology, captured by cosine affinities between class prototypes, as a stable cross-domain structure under diverse 3D corruptions. This observation establishes class topology as a robust source of domain-shared information for continual adaptation.
- We propose ToCoTTA, which leverages class topology in three complementary roles: a **Prior** that refines online prototypes, a **Monitor** that assesses adaptation reliability, and a **Repair** that mitigates error accumulation and catastrophic forgetting through adaptive restoration.
- ToCoTTA achieves state-of-the-art performance across multiple models and datasets under continual adaptation settings. It further shows consistent improvements under long-horizon adaptation while incurring negligible computational overhead.

2 Related Work

2.1 Test-Time Adaptation

Test-time adaptation (TTA) adapts a source-trained model to unseen target distributions using only unlabeled target data at inference time. Most TTA methods (Wang et al., 2021; Zhang et al., 2022) employ self-supervised objectives such as entropy minimization and often update only a subset of parameters, typically normalization layers (Osowiechi et al., 2024; Maharana et al., 2025; Bao et al., 2025) or learnable prompts (Shu et al., 2022; Feng et al., 2023; Yoon et al., 2024; Ahamed et al., 2026; Fillioux et al., 2026; Jiang et al., 2026). Recently, cache-based methods (Karmanov et al., 2024; Zhang et al., 2024; Chen et al., 2025) have been explored for efficient adaptation. Beyond discrete cache-based methods, several approaches (Fan et al., 2025; Han et al., 2025; Zhang et al., 2025) construct continuous caches to better capture the target distribution. Along this direction, recent methods (Xiao et al., 2026; Murphy et al., 2026) explore backpropagation-free adaptation to improve test-time efficiency. TTA has also been extended to 3D point clouds, with recent methods focusing on efficient adaptation (Yazdanpanah et al., 2025; 2026), while SVWA (Bahri et al., 2025) improves performance through weight averaging.

2.2 Continual Test-Time Adaptation

Continual test-time adaptation (CTTA) extends TTA to continuously evolving target distributions, where the model adapts without access to source data, target labels, or domain-boundary information. CoTTA (Wang et al., 2022) mitigates error accumulation and catastrophic forgetting through a mean–teacher framework and stochastic restoration, while EATA (Niu et al., 2022; Tan et al., 2025) regularizes important parameters using Fisher information to alleviate forgetting. Subsequent methods improve robustness and efficiency through instance-aware normalization and memory sampling (Gong et al., 2022), lightweight meta networks with self-distilled regularization (Song et al., 2023), or domain-adaptive low-rank experts (Lee et al., 2024). Distribution-aware approaches (Yang et al., 2024; Liu et al., 2024a;b) leverage signals of distributional change to guide adaptation through different mechanisms, including visual prompts (Gan et al., 2023; Yang et al., 2024), masked image modeling with HOG features (Liu et al., 2024a), and lightweight adapters (Liu et al., 2024b). More recently, several methods (Yoo et al., 2024; Zhao et al., 2026; Park et al., 2025) explicitly detect domain shifts and adjust adaptation accordingly. Despite their effectiveness, most CTTA methods rely on teacher—student frameworks for stable adaptation, introducing additional memory and inference overhead. Domain shift detection can further increase computational cost by requiring additional mechanisms to estimate or identify distributional changes. More fundamentally, partitioning the test stream according to distribution shifts may overlook useful knowledge shared across domains.

3 Class Topology Survives Corruption

Problem Definition. Continual test-time adaptation (CTTA) aims to adapt a pre-trained model to continuously changing target domains during inference. Given a model f_{θ_0} trained on source data $D_S = (X_S, Y_S)$, the goal is to improve its performance over a sequence of target domains $D_{T_1}, D_{T_2}, \ldots, D_{T_n}$, where each target dataset $D_{T_i} = (X_{T_i})$ consists of unlabeled test samples and n denotes the number of target domains. Throughout adaptation, the source data are inaccessible. Moreover, unlabeled target samples arrive sequentially in an online manner, with each sample accessible only once. CTTA must adapt solely from unlabeled target samples, without prior knowledge of when domain shifts occur. This online self-training process, typically based on pseudo-labels, makes CTTA susceptible to catastrophic forgetting and error accumulation over time.

Recent CTTA methods often leverage distribution-shift information, such as shift magnitude or domain transitions, to inform the adaptation process. However, such strategies primarily emphasize differences across domains and may overlook knowledge shared across them. Instead, we investigate what remains stable across changing target domains. We observe that, despite substantial corruption-induced shifts, the underlying relational structure among classes, which we refer to as *class topology*, remains stable.

We define class topology based on the pairwise cosine affinities between class prototypes in the feature space. Let ϕ denote the encoder of the model, which maps an input x to its penultimate

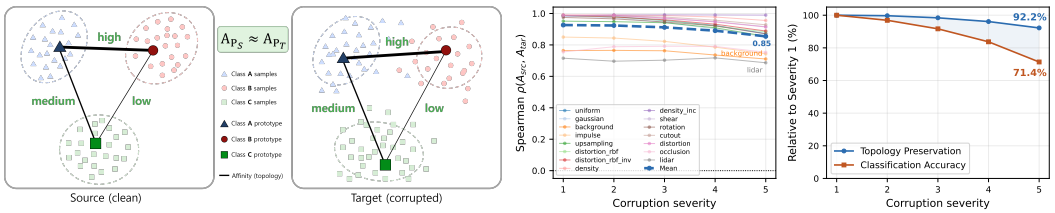


Figure 2: (**Left**) Although corruption shifts and overlaps class feature distributions, the relative geometry among class prototypes remains largely preserved. (**Right**) Across diverse corruption types and severities, prototype-affinity topology remains highly correlated with its clean counterpart and degrades much more slowly than classification accuracy.

representation $z = \phi(x) \in \mathbb{R}^d$ before the final linear classifier h. For each class c, we first compute its prototype $\mathbf{p}^c$ as the mean feature of the samples belonging to that class as follows:

$$\mathbf{p}^c = \frac{1}{|D^c|} \sum_{(x,y)\in D^c} z, \qquad c = 1, 2, \ldots, C, \tag{1}$$

where $D^c = \{(x, y) \in D \mid y = c\}$ denotes the subset of samples in D belonging to class c. We ℓ_2-normalize both the features and class prototypes. Given the set of class prototypes $\mathbf{P} \in \mathbb{R}^{C\times d}$, we define the class topology as a cosine affinity matrix $\mathbf{A}_\mathbf{P} \in \mathbb{R}^{C\times C}$ as follows:

$$\mathbf{A}_\mathbf{P}^{ij} = \cos\big(\mathbf{p}^i, \mathbf{p}^j\big) = \frac{\mathbf{p}^{i\top}\mathbf{p}^j}{\|\mathbf{p}^i\|_2 \|\mathbf{p}^j\|_2}, \tag{2}$$

where $\mathbf{A}_\mathbf{P}^{ij} \in [-1, 1]$, with larger values indicating greater affinity between classes i and j.

We compare the affinity matrix of each target domain $\mathbf{A}_{\mathbf{P}_{T_i}}$ with the source (clean) affinity matrix $\mathbf{A}_{\mathbf{P}_S}$ using Spearman rank correlation. As shown in Fig. 2, class topology remains highly consistent across diverse corruption types and severity levels, with the mean correlation remaining at 0.85 even at the highest severity. Moreover, it degrades substantially more slowly than classification accuracy as corruption severity increases. These results indicate that the relative relationships among classes are largely preserved, motivating the use of the source topology as a stable anchor.

Source-free Topology Reference. However, computing $\mathbf{A}_{\mathbf{P}_S}$ requires access to the source data, which are unavailable at test time. We approximate the source topology from the classifier weights. Let $W \in \mathbb{R}^{C\times d}$ denote the weight matrix of the final linear classifier h, where each row corresponds to a class. We observe that the affinity matrix of the ℓ_2-normalized classifier weights $\hat{W}$ closely matches the source affinity, achieving a Spearman rank correlation of $\rho = 0.90$. We therefore define the reference affinity as $\mathbf{A}_{\text{ref}} = \mathbf{A}_{\hat{W}}$. This provides a source-free topology reference that requires no access to source data, as it is already encoded in the pre-trained classifier.

4 ToCoTTA: One Topology, Three Roles

To leverage this reference topology throughout continual adaptation, we introduce ToCoTTA, which incorporates an auxiliary prototype head and uses reference topology in three complementary roles: **prior**, **monitor**, and **repair**. For incoming test samples, an online prototype bank first accumulates class-wise features. The reference topology then serves as a **prior** to refine these prototypes into class anchors, which are used for both prediction and adaptation (§4.1). It also acts as a **monitor** that measures the reliability of the evolving target topology (§4.2). The resulting trust signal controls **repair** mechanisms that stabilize adaptation (§4.3). The overall framework is illustrated in Fig. 3.

Online Prototype Bank. To estimate the topology of the evolving target distribution, we construct class-wise prototypes summarizing its feature distribution. Specifically, we maintain an online prototype bank $\mu \in \mathbb{R}^{C\times d}$ using soft assignments and exponential decay. Given a batch $\mathcal{B}$ and the corresponding classifier probabilities $p(x) = \text{softmax}(h(z))$, we update each class prototype as:

$$\mu_c \leftarrow \frac{\gamma\, s_c\, \mu_c + \sum_{x\in\mathcal{B}} p_c(x)\, z}{\gamma\, s_c + \sum_{x\in\mathcal{B}} p_c(x)}, \qquad s_c \leftarrow \gamma\, s_c + \textstyle\sum_{x\in\mathcal{B}} p_c(x). \tag{3}$$

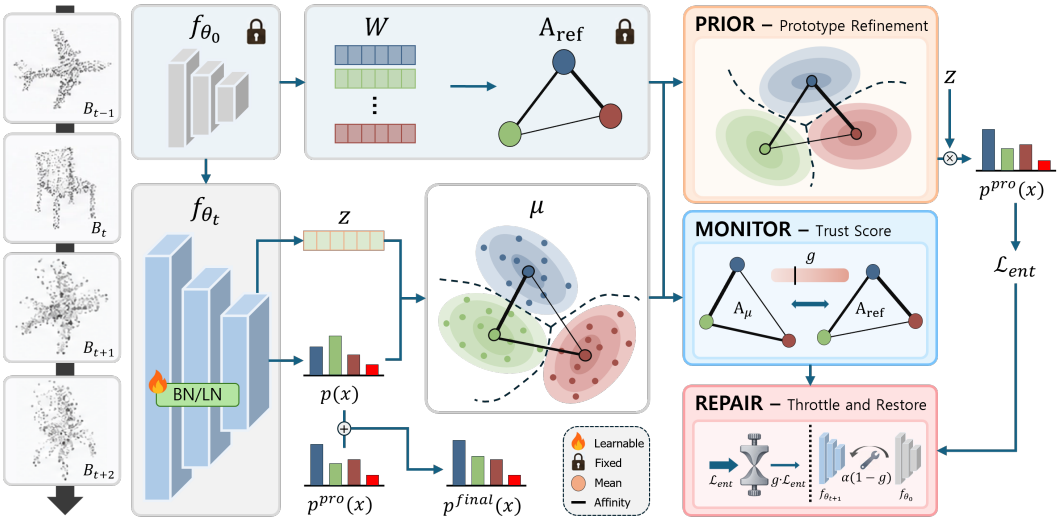


Figure 3: The reference topology $\mathbf{A}_{\mathrm{ref}}$ is extracted from fixed classifier weights and reused throughout continual adaptation. As a **Prior**, it refines online prototypes for prediction and adaptation. As a **Monitor**, it measures alignment with the evolving target topology to estimate a trust score g. As a **Repair** mechanism, g throttles unreliable updates and controls the degree of parameter restoration.

Here, s_c denotes the accumulated assignment weight for class c, and γ controls exponential decay. The exponential decay gradually suppresses erroneous contributions from previous domains. It also downweights unreliable features from early adaptation stages, thereby placing more emphasis on recent features. We initialize μ and s from the initial batch.

4.1 Prior: Topology-Guided Prototype Refinement

The online prototype bank μ captures the evolving class-wise feature structure, but its pairwise class relations can be distorted by noisy pseudo-labels and accumulated errors. The reference topology provides a stable prior for these inter-class relations. We therefore refine the target prototypes using this topology prior. Specifically, we obtain the refined prototypes $\mathbf{P}^\star$ by solving:

$$\mathbf{P}^\star = \arg\min_{\mathbf{P}} \underbrace{\|\mathbf{P}-\mu\|_F^2}_{\text{data fidelity}} + \lambda \underbrace{\mathrm{KL}\big(\sigma(\mathbf{A}_{\mathrm{ref}}) \,\|\, \sigma(\mathbf{A}_{\mathbf{P}})\big)}_{\text{topology prior}}, \tag{4}$$

where $\sigma(\cdot)$ denotes row-wise softmax over off-diagonal affinities, and KL is computed row-wise. The first term preserves the target prototypes captured by μ, while the second aligns their pairwise relations with the reference topology. Importantly, this topology prior constrains only the relative class structure rather than anchoring the prototypes to their source positions, allowing them to follow the target distribution while preserving inter-class relations. The refined prototypes $\mathbf{P}^\star$ are then used as class anchors for prototype-based prediction and adaptation, as described in the following section. Since the prototype bank evolves gradually across batches, we warm-start the refinement from the previous solution and perform a single optimization step per batch.

4.1.1 Prediction and Adaptation

The refined prototypes serve as class anchors for both prediction and adaptation. Given an input x and its feature z, we construct an auxiliary prototype head based on cosine similarities to the refined anchors $\mathbf{P}^\star$ and combine its prediction with that of the base head as follows:

$$p^{\mathrm{final}}(x) = p(x) + p^{\mathrm{pro}}(x), \qquad p^{\mathrm{pro}}(x) = \mathrm{softmax}\left(\cos\left(\mathbf{P}^\star, z\right)\right). \tag{5}$$

We adapt the model using entropy minimization (Wang et al., 2021) on the prototype head as:

$$\mathcal{L}_{\mathrm{ent}} = \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \mathcal{H}\left(p^{\mathrm{pro}}(x)\right). \tag{6}$$

This objective pulls target features toward topology-guided class anchors, thereby directly incorporating the reference topology into adaptation. However, erroneous predictions can still reinforce themselves through self-training, highlighting the need to assess adaptation reliability.

4.2 MONITOR: TOPOLOGY-BASED TRUST SCORE

The reference topology also provides a natural criterion for assessing the reliability of the evolving target representation. We measure this reliability directly from the online prototype bank μ by comparing its topology with the reference topology as follows:

$$g = \mathrm{clamp}\left(\frac{r - t_0}{t_1 - t_0}, 0, 1\right), \qquad r = \mathrm{corr}\big(\mathbf{A}_\mu, \mathbf{A}_{\mathrm{ref}}\big) \tag{7}$$

where r denotes the Pearson correlation between off-diagonal affinities, and $g \in [0, 1]$ is the resulting topology-based trust score. A high g indicates that the current inter-class relations remain consistent with the reference topology, whereas a low g signals structural degradation in the prototype bank. We compute the trust score from the raw prototype bank μ rather than the refined prototypes $\mathbf{P}^\star$. Since $\mathbf{P}^\star$ is optimized toward $\mathbf{A}_{\mathrm{ref}}$, it may overestimate reliability, whereas μ reflects the target structure before refinement. The resulting trust score is then used to control adaptation.

4.3 REPAIR: THROTTLING AND RESTORATION

We use the trust score in two complementary ways to prevent further error accumulation and recover from accumulated parameter errors. **(i) Throttling.** We scale the adaptation loss in Eq. 6 by the trust score, *i.e.*, $\mathcal{L}_{\mathrm{ent}} = g \cdot \mathcal{L}_{\mathrm{ent}}$, so that self-training is adaptively suppressed when the target topology becomes unreliable. **(ii) Restoration.** Throttling alone can prevent additional drift, but it cannot undo errors already accumulated in the adapted parameters. We therefore restore the trainable parameters toward their source values as follows:

$$\theta_t^a \leftarrow (1 - k) \cdot \theta_t^a + k \cdot \theta_0^a, \qquad k = \alpha(1 - g), \tag{8}$$

where θ_t^a denotes the trainable affine parameters of the normalization layers at time step t, and θ_0^a denotes their source values. The coefficient α controls the maximum restoration strength. As trust decreases, restoration becomes stronger, pulling the model back toward the source state. By keeping the adapted model from drifting too far from its source state, restoration also helps mitigate catastrophic forgetting. Together, throttling limits further error accumulation, while restoration reduces drift already accumulated in the model parameters.

5 EXPERIMENT

Datasets. We evaluate ToCoTTA on three 3D point cloud classification datasets: ModelNet40 (MN) (Wu et al., 2015), ScanObjectNN (SONN) (Uy et al., 2019), and ShapeNet (SN) (Chang et al., 2015). MN and SN are synthetic datasets with 40 and 55 classes, respectively, while SONN contains 15 classes of real-world scanned objects. We use ModelNet40-C (MN-C) (Sun et al., 2022), which includes 15 corruption types, and construct SONN-C and SN-C using the same corruptions, each with five severity levels. Unless otherwise specified, we use severity level 5.

Implementation Details. We use three 3D point cloud classification backbones as source models: Point-MAE (Pang et al., 2022), DGCNN (Wang et al., 2019), and PointNet (Qi et al., 2017). Each model is pre-trained on its respective clean dataset and subsequently adapted to the corrupted test data. For all model–dataset combinations, we use the same hyperparameters: a batch size of 32, a learning rate of 0.001, $\gamma = 0.98$, $\lambda = 10$, $t_0 = 0.4$, $t_1 = 0.8$, and $\alpha = 0.05$. During adaptation, only the normalization layers are updated, using a single optimization step for each test batch.

5.1 MAIN RESULTS

ModelNet40-C. Table 1 reports results on MN-C across three backbones. The upper, middle, and lower groups correspond to backpropagation-free, learning-based TTA, and CTTA methods, respectively. ToCoTTA consistently achieves the state-of-the-art average accuracy across all three

Table 1: Classification accuracy (%) on ModelNet40-C across corruption types and backbones. Best and second-best results are highlighted in **bold** and underline, respectively.

	Time	t →															
	Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean
Point-MAE	Source-Only	66.57	59.16	7.01	31.73	65.56	67.71	69.81	77.96	85.25	74.35	38.09	74.76	70.02	38.53	30.47	57.13
	T3A	72.69	70.02	9.24	44.77	71.64	72.53	74.47	80.79	85.83	78.93	43.23	78.93	74.39	37.72	31.48	61.78
	DUA	79.62	77.88	17.22	68.27	63.09	72.73	76.30	84.08	85.39	78.85	52.67	80.63	76.58	41.82	40.80	66.39
	Purge-Gate	78.12	74.47	**40.88**	66.69	81.73	73.58	75.81	79.09	82.37	77.23	58.19	75.49	73.58	47.85	45.14	68.68
	LN-TTA	79.86	75.53	29.30	67.99	81.81	76.01	76.42	80.88	83.47	78.32	60.17	77.59	74.88	52.47	51.42	69.74
	PL	75.85	71.52	23.50	64.14	73.38	73.70	75.28	79.82	83.91	74.76	56.93	76.90	70.54	45.75	45.18	66.08
	TENT	75.73	72.08	22.93	64.87	73.66	73.78	75.73	80.23	83.71	75.53	58.27	77.27	70.46	46.56	45.87	66.45
	SHOT	75.53	75.85	29.35	70.30	74.47	75.89	74.11	77.43	78.28	67.38	54.38	71.43	65.36	46.03	43.23	65.27
	MEMO	63.01	54.78	4.82	23.30	65.15	66.25	68.19	78.32	85.45	74.51	36.35	74.15	69.77	38.41	29.21	55.45
	SVWA	79.78	75.81	25.61	66.90	75.00	75.24	77.51	82.66	85.82	79.94	60.37	81.28	76.99	51.34	48.87	69.54
	CoTTA	80.21	78.66	29.15	69.83	78.15	78.38	78.84	83.68	85.28	81.69	66.92	81.93	78.55	53.97	55.16	72.03
	ViDA	79.82	76.99	25.53	67.91	76.34	77.07	77.51	83.02	86.18	80.51	63.29	81.85	78.36	51.26	51.22	70.46
	Hybrid-TTA	80.07	76.62	25.28	67.42	82.70	77.96	78.53	81.08	84.32	82.33	64.26	80.11	78.97	53.61	55.07	71.22
	ToCoTTA	**81.52**	**79.09**	30.62	**74.60**	**86.10**	**80.19**	**82.09**	**86.26**	**86.41**	**82.87**	**68.36**	**82.85**	**80.23**	**55.81**	**58.71**	**74.38**
DGCNN	CoTTA	80.47	81.81	44.13	75.89	80.51	81.16	80.88	79.84	84.37	82.42	76.84	81.77	80.04	45.46	40.59	73.08
	ViDA	80.51	82.78	36.10	75.20	81.16	81.44	81.73	79.07	85.31	82.70	75.12	82.13	80.85	39.51	37.68	72.09
	ToCoTTA	**82.25**	**84.08**	**46.15**	**79.09**	**82.74**	**83.10**	**84.36**	**82.46**	**87.20**	**84.20**	**78.44**	**83.51**	**82.13**	**49.23**	**46.84**	**75.72**
PointNet	CoTTA	70.56	82.62	12.88	67.26	71.68	64.59	66.37	83.55	84.48	51.70	41.01	81.85	58.59	36.47	40.33	60.93
	ViDA	70.02	82.17	11.75	64.18	70.02	61.95	64.22	82.82	84.12	45.46	33.91	81.20	53.12	28.20	29.09	57.48
	ToCoTTA	**71.27**	**83.83**	**13.53**	**69.57**	**74.35**	**67.30**	**69.33**	**84.64**	**84.93**	**54.13**	**45.95**	**83.06**	**63.25**	**37.20**	**41.29**	**62.91**

Table 2: Classification accuracy (%) on ScanObjectNN-C under diverse corruption types.

	Time	t →															
	Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean
Point-MAE	Source-Only	15.66	15.15	12.74	15.84	14.11	40.45	41.48	70.74	70.05	40.45	29.60	69.02	42.69	6.20	8.09	32.82
	T3A	17.56	18.76	15.32	18.59	15.66	44.41	48.37	73.32	73.15	41.31	32.19	71.60	46.13	6.37	9.12	35.46
	DUA	24.61	36.66	10.84	30.47	28.23	44.41	46.99	61.96	70.05	46.47	38.38	65.06	51.46	7.75	10.16	38.23
	Purge-Gate	40.96	54.73	21.86	36.49	39.59	55.25	58.35	75.56	74.01	56.80	48.71	75.56	58.69	7.06	9.47	47.54
	LN-TTA	43.20	57.49	**25.13**	40.28	43.37	56.63	60.59	76.25	74.01	56.28	50.95	77.63	60.59	7.40	9.81	49.31
	PL	30.98	44.58	12.91	32.19	31.67	50.09	53.70	71.77	69.88	50.78	45.96	74.01	54.56	7.40	8.95	42.63
	TENT	30.29	45.96	13.08	32.70	33.05	49.40	52.50	73.67	71.60	50.60	45.96	74.36	54.91	7.57	8.78	42.96
	SHOT	34.08	52.67	12.39	37.69	39.59	51.81	57.66	73.84	73.67	57.14	50.78	75.04	60.41	6.20	8.26	46.08
	MEMO	14.80	14.11	11.70	13.08	12.39	34.42	36.83	64.54	68.85	37.01	27.02	65.40	36.49	7.06	8.78	30.17
	SVWA	33.22	49.23	14.11	36.83	34.25	53.36	55.25	75.56	74.11	54.91	47.33	74.01	56.28	8.26	9.64	45.09
	CoTTA	37.87	51.64	13.08	36.66	35.97	54.39	56.45	73.32	72.98	57.66	49.40	74.18	57.14	7.92	9.47	45.87
	ViDA	33.91	45.78	13.43	36.83	33.39	53.36	53.87	75.22	72.98	54.91	46.82	74.18	56.45	8.26	10.12	44.63
	Hybrid-TTA	38.21	54.91	13.25	32.87	38.90	53.53	57.14	75.39	71.26	57.32	46.64	75.90	58.69	8.95	10.50	46.23
	ToCoTTA	**43.82**	**59.38**	23.00	**41.31**	**44.41**	**57.45**	**60.62**	**76.40**	**75.90**	**59.40**	**52.50**	**77.73**	**61.45**	**9.43**	**10.84**	**50.24**

Table 3: Classification accuracy (%) on ShapeNet-C under diverse corruption types.

Method	Source	T3A	DUA	PG	LN-TTA	PL	TENT	SHOT	MEMO	SVWA	CoTTA	ViDA	HTTA	ToCoTTA
Point-MAE	59.98	58.69	62.00	64.40	64.37	60.66	60.97	28.18	60.13	63.36	64.98	63.71	66.12	**68.56**

backbones. With Point-MAE, ToCoTTA reaches a mean accuracy of 74.38%, outperforming the strongest baseline, CoTTA, by 2.35%. This trend also holds for DGCNN and PointNet, where ToCoTTA achieves 75.72% and 62.91%, respectively. Moreover, ToCoTTA achieves the best performance on all but one corruption across the three backbones, demonstrating consistent robustness to diverse corruptions rather than gains limited to a particular corruption type or architecture.

ScanObjectNN-C. Table 2 presents results on SONN-C, which consists of real-world scanned objects and therefore exhibits more challenging geometric variations. ToCoTTA achieves the highest mean accuracy of 50.24% and performs best on most corruption types, highlighting its effectiveness under realistic scanning artifacts and severe distribution shifts.

ShapeNet-C. As shown in Tab. 3, ToCoTTA also consistently improves performance on SN-C. It achieves a mean accuracy of 68.56%, surpassing the strongest baseline, Hybrid-TTA, by 2.44%. Notably, ToCoTTA achieves the highest accuracy on all 15 corruption types, indicating that its advantage generalizes to a dataset with a larger number of object categories and diverse shape structures. The full results are provided in the Appendix (§A.5).

Long-Horizon Continual Adaptation. We further evaluate the long-horizon stability of ToCoTTA by adapting the model over the corruption sequence for 10 consecutive rounds without

Table 4: Long-horizon adaptation on ModelNet-C over 10 corruption rounds without model reset.

Round	R →										
Method	1	2	3	4	5	6	7	8	9	10	Mean
CoTTA	72.03	72.93	72.81	73.39	73.20	73.23	73.67	73.27	73.79	74.08	73.24
ViDA	70.46	71.51	71.77	72.39	72.78	73.00	73.32	73.09	73.32	73.89	72.55
Hybrid-TTA	71.22	71.55	72.55	73.16	73.14	73.07	73.42	73.61	73.57	73.47	72.88
ToCoTTA	**74.38**	**75.00**	**75.33**	**75.56**	**75.59**	**76.03**	**76.31**	**76.16**	**75.83**	**76.34**	**75.65**

Table 5: Comparison of different topology prior designs.

$R(\mathbf{P})$	SF	MN-C	SONN-C	SN-C
None	✓	72.29	48.91	66.42
$\|\mathbf{P}-\hat{W}\|_F^2$	✓	72.19	48.42	66.74
$\mathrm{KL}(\mathbf{A}_{\mathbf{P}_S}\|\mathbf{A}_{\mathbf{P}})$	✗	**74.61**	**50.42**	**68.65**
$\mathrm{KL}(\mathbf{A}_{\mathrm{ref}}\|\mathbf{A}_{\mathbf{P}})$	✓	74.38	50.24	68.56

Table 6: Ablation study of individual components.

$\mathcal{L}_{ent}$	p^{pro}	g^T	g^R	MN-C	SONN-C	SN-C
Base	✓	✓	✓	72.37	48.48	66.27
Pro.		✓	✓	73.92	49.79	68.05
Pro.	✓			70.51	48.02	65.94
Pro.	✓	✓		71.15	48.54	66.15
Pro.	✓		✓	73.93	49.97	67.91
Pro.	✓	✓	✓	**74.38**	**50.24**	**68.56**

resetting the model. As shown in Tab. 4, ToCoTTA consistently maintains higher accuracy throughout all rounds and achieves a mean accuracy of 75.65%, outperforming all existing CTTA methods. These results demonstrate that ToCoTTA remains stable under prolonged continual adaptation without substantial error accumulation or performance degradation.

5.2 Ablation Study

5.2.1 Effect of Three Roles

Topology Prior Design. We analyze the effect of the topology prior by formulating prototype refinement as $\mathbf{P}^\star = \arg\min_{\mathbf{P}} \|\mathbf{P}-\mu\|_F^2 + \lambda R(\mathbf{P})$, where $R(\mathbf{P})$ specifies the prior imposed on the prototypes. As shown in Tab. 5, preserving the relative structure among classes consistently outperforms both using no prior and directly anchoring the prototypes to the source classifier weights, while performing comparably to an oracle topology constructed from source prototypes without requiring source data. The first two rows of Tab. 6 further show how the refined prototypes contribute to both prediction and adaptation. Fusing p^{pro} improves performance, and applying $\mathcal{L}_{ent}$ to prototype head instead of the conventional base head yields a further improvement. These results suggest that topology-guided prototypes are effective both as prediction anchors and as optimization targets.

Topology-Based Reliability Monitor. We evaluate whether the topology-based trust score g provides a reliable indicator of adaptation quality. As shown in Fig. 4 (left), g closely tracks the temporal variation in accuracy as the model encounters successive corruption domains, decreasing under challenging conditions and recovering as prediction accuracy improves. Figure 4 (right) further shows a strong positive correlation between g and accuracy across corruption types, indicating that consistency with the reference topology serves as an effective proxy for model reliability without access to target labels. Table 6 also shows the practical utility of this reliability signal, where w/o g^T and w/o g^R denote disabling topology-based control for throttling and restoration, respectively. Using g to adaptively control both mechanisms consistently improves accuracy, demonstrating the benefit of reliability-aware repair. Together, these results show that the proposed monitor reliably reflects adaptation reliability and provides an effective signal for controlling continual adaptation.

Controlling Repair. Table 7 analyzes the complementary roles of g^T, g^R, and restoration ("Rest."). g^T suppresses unreliable updates, while g^R controls restoration strength. Comparing the second row with fourth row of Tab. 6 shows that restoration should be applied selectively. On MN-C and SONN-C, removing restoration causes minor degradation, whereas removing g^R leads to a substantial performance drop. In contrast, SN-C benefits substantially from restoration. Thus, g^R plays a key role in triggering restoration only when necessary. The bottom part further highlights the importance of repair under long-horizon adaptation. Compared with the single-round setting, removing each component causes larger degradation. Notably, removing only g^T while retaining

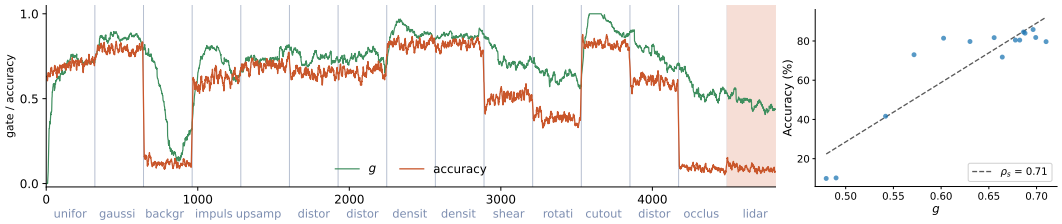


Figure 4: Analysis of topology-based reliability monitor. (**Left**) Temporal evolution of g and classification accuracy. (**Right**) Correlation between g and classification accuracy across 15 corruptions.

Table 7: Effect of each component on repair.

Method	MN-C	SONN-C	SN-C
ToCoTTA	74.38	50.24	68.56
− Rest.	73.53	49.93	63.38
$-g^T$ − Rest.	73.36	49.70	60.90
ToCoTTA × 10R	75.65	50.99	68.64
$-g^T$	74.41	49.84	67.51
$-g^R$	70.01	48.55	66.05
− Rest.	66.29	48.36	57.41
$-g^T$ − Rest.	58.57	46.71	12.33

Table 8: Efficiency and accuracy comparison.

Method	Throughput ↑ (samples/s)	Cost ↓	Acc ↑ (%)
Source-only	1,907	×1	57.13
TENT	**751**	×**2.5**	66.45
SVWA	79	×24.1	69.54
CoTTA	32	×59.6	<u>72.03</u>
ViDA	92	×20.7	70.46
Hybrid-TTA	293	×6.5	71.22
ToCoTTA	<u>685</u>	<u>×2.8</u>	**74.38**

restoration causes moderate degradation, whereas removing both leads to severe performance collapse (last row). This indicates that g^T primarily prevents unreliable updates from accumulating over time and becomes particularly important when restoration is unavailable.

Efficiency. Table 8 compares the computational efficiency and accuracy of different methods. ToCoTTA achieves 685 samples/s at a computational cost of only ×2.8, while attaining the highest accuracy of 74.38%. Compared with CoTTA, ToCoTTA provides over 21× higher throughput while improving accuracy by 2.35 percentage points. It also achieves substantially higher throughput than ViDA and Hybrid-TTA, demonstrating a favorable efficiency–accuracy trade-off.

6 Limitation and Future Work

Several aspects of ToCoTTA remain open to further investigation. $\mathbf{A}_{\text{ref}}$ is derived from fixed classifier weights, which serve as a proxy for the source-domain class structure. However, these weights may not fully reflect the relational structure of the source feature space. Although this design enables source-free adaptation, developing more faithful source-free estimates of class topology is an important direction for future work. We also evaluate ToCoTTA only on classification tasks. Extending it beyond classification and evaluating it under non-IID or dynamic test streams remain natural directions. More broadly, our results suggest that class topology is only one form of domain-shared information. Exploring other transferable signals may further improve continual adaptation.

7 Conclusion

We present ToCoTTA, a topology-anchored framework for continual test-time adaptation of 3D point clouds. Our key finding is that class topology, defined by the relational structure among class prototypes, remains remarkably stable even when corruption substantially degrades classification performance. Building on this observation, ToCoTTA uses a single reference class topology as a prior for prototype refinement, a monitor for adaptation reliability assessment, and a repair for mitigating error accumulation. Across multiple datasets and backbones, ToCoTTA consistently outperforms existing TTA and CTTA methods while preserving long-horizon stability and computational efficiency. These results suggest that explicitly identifying and leveraging domain-shared information provides a promising direction for robust continual adaptation under evolving test distributions.

AI USE STATEMENT

In this work, we used generative AI tools for designing or providing feedback on research methodology or experiments, and assisting with translation. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, implementing methods, and interpreting results, and generating synthetic datasets, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, cleaning and reformatting dataset, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, suggesting experimental parameters, creating or editing software code, creation of artifacts, editing a research paper to improve readability. We have reviewed all AI-assisted work. LLM-generated language edits were manually reviewed and approved by the authors before being incorporated into the manuscript, experimental suggestions were independently evaluated by the authors, and LLM-generated code was verified and tested for correctness before use. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide implementation details, evaluation protocols, and hyperparameter settings in Sec. 5. A more detailed experimental setup, including hardware specifications, and additional sensitivity analyses are provided in Appendices A.1 and A.4, respectively. All datasets used in our experiments are publicly available. Anonymous source code is provided in the supplementary material to facilitate reproduction of our results.

REFERENCES

Shihab Aaqil Ahamed, Udaya Sampath K. Perera Miriya Thanthrige, Ranga Rodrigo, and Muhammad Haris Khan. A-TPT: Angular diversity calibration properties for test-time prompt tuning of vision-language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ali Bahri, Moslem Yazdanpanah, Mehrdad Noori, Sahar Dastani Oghani, Milad Cheraghalikhani, David Osowiechi, Farzad Beizaee, Gustavo A. Vargas Hakim, Ismail Ben Ayed, and Christian Desrosiers. Test-time adaptation in point clouds: Leveraging sampling variation with weight averaging. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 266–275, 2025.

Wenxuan Bao, Ruxi Deng, and Jingrui He. Mint: A simple test-time adaptation of vision-language models against common corruptions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository, 2015.

Xinyu Chen, Haotian Zhai, Can Zhang, Xiupeng Shi, and Ruirui Li. Multi-cache enhanced prototype learning for test-time generalization of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2281–2291, October 2025.

Xinqi Fan, Xueli Chen, Luoxiao Yang, Chuin Hong Yap, Rizwan Qureshi, Qi Dou, Moi Hoon Yap, and Mubarak Shah. Test-time retrieval-augmented adaptation for vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8810–8819, October 2025.

Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2704–2714, October 2023.

Leo Fillioux, Omprakash Chakraborty, Ismail Ben Ayed, Paul-Henry Cournède, Stergios Christodoulidis, Maria Vakalopoulou, and Jose Dolz. Soc: Semantic orthogonal calibration for

test-time prompt tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4772–4782, June 2026.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059. PMLR, 20–22 Jun 2016.

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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):7595–7603, 2023.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. NOTE: Robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems*, 2022.

Zongbo Han, Jialong Yang, Guangyu Wang, Junfan Li, Qianli Xu, Mike Zheng Shou, and Changqing Zhang. DOTA: Distributional test-time adaptation of vision-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Siru Jiang, Jian Liang, Ran He, and Tieniu Tan. USE: A unified self-ensembling framework for test-time prompt tuning. In *Forty-third International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 6–11 Jul 2026.

Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14162–14171, June 2024.

Hyeongyu Kim, GeonHui Han, and Dosik Hwang. Buffer layers for test-time adaptation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Daeun Lee, Jaehong Yoon, and Sung Ju Hwang. BECoTTA: Input-dependent online blending of experts for continual test-time adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 27072–27093. PMLR, 21–27 Jul 2024.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896, 2013.

Jiaming Liu, Ran Xu, Senqiao Yang, Renrui Zhang, Qizhe Zhang, Zehui Chen, Yandong Guo, and Shanghang Zhang. Continual-mae: Adaptive distribution masked autoencoders for continual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28653–28663, June 2024a.

Jiaming Liu, Senqiao Yang, Peidong Jia, Renrui Zhang, Ming Lu, Yandong Guo, Wei Xue, and Shanghang Zhang. ViDA: Homeostatic visual domain adapter for continual test time adaptation. In *The Twelfth International Conference on Learning Representations*, 2024b.

Sarthak Maharana, Baoming Zhang, Leonid Karlinsky, Rogerio Feris, and Yunhui Guo. Batclip: Bimodal online test-time adaptation for clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1569–1579, October 2025.

Alexander Murphy, Michal Danilowski, Soumyajit Chatterjee, and Abhirup Ghosh. NEO — no-optimization test-time adaptation through latent re-centering. In *The Fourteenth International Conference on Learning Representations*, 2026.

Chenggong Ni, Fan Lyu, Jiayao Tan, Fuyuan Hu, Rui Yao, and Tao Zhou. Maintaining consistent inter-class topology in continual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15319–15328, June 2025.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905. PMLR, 17–23 Jul 2022.

David Osowiechi, Mehrdad Noori, Gustavo Adolfo Vargas Hakim, Moslem Yazdanpanah, Ali Bahri, Milad Cheraghalikhani, Sahar Dastani, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. WATT: Weight average test time adaptation of CLIP. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision (ECCV)*, pp. 604–621, 2022.

Hyewon Park, Hyejin Park, Jueun Ko, and Dongbo Min. Hybrid-tta: Continual test-time adaptation via dynamic domain shift detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2877–2886, October 2025.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *Advances in Neural Information Processing Systems*, 2022.

Junha Song, Jungsoo Lee, In So Kweon, and Sungha Choi. Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11920–11929, June 2023.

Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, Chaowei Xiao, and Z. Morley Mao. Benchmarking robustness of 3d point cloud recognition against common corruptions, 2022.

Mingkui Tan, Guohao Chen, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Peilin Zhao, and Shuaicheng Niu. Uncertainty-calibrated test-time model adaptation without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6274–6289, 2025.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, 2017.

Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

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

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.*, 38(5), October 2019.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

MA Xiao, Young D. Kwon, Pan Zhou, and Dong Ma. Architecture-agnostic test-time adaptation via backprop-free embedding alignment. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zehao Xiao, Shilin Yan, Jack Hong, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiayi Shen, Cheems Wang, and Cees G. M. Snoek. Dynaprompt: Dynamic test-time prompt tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Senqiao Yang, Jiarui Wu, Jiaming Liu, Xiaoqi Li, Qizhe Zhang, Mingjie Pan, Yulu Gan, Zehui Chen, and Shanghang Zhang. Exploring sparse visual prompt for domain adaptive dense prediction, 2024.

Moslem Yazdanpanah, Ali Bahri, Mehrdad Noori, Sahar Dastani, Gustavo Adolfo Vargas Hakim, David Osowiechi, Ismail Ben Ayed, and Christian Desrosiers. Purge-gate: Backpropagation-free test-time adaptation for point clouds classification via token purging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27640–27649, October 2025.

Moslem Yazdanpanah, Ali Bahri, Mehrdad Noori, Sahar Dastani, Samuel Barbeau, David Osowiechi, Gustavo Adolfo Vargas Hakim, Ismail Ben Ayed, and Christian Desrosiers. Revisiting layer normalization for point cloud test time adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 537–546, March 2026.

Jayeon Yoo, Dongkwan Lee, Inseop Chung, Donghyun Kim, and Nojun Kwak. What how and when should object detectors update in continually changing test domains? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23354–23363, June 2024.

Hee Suk Yoon, Eunseop Yoon, Joshua Tian Jin Tee, Mark A. Hasegawa-Johnson, Yingzhen Li, and Chang D. Yoo. C-TPT: Calibrated test-time prompt tuning for vision-language models via text feature dispersion. In *The Twelfth International Conference on Learning Representations*, 2024.

Ce Zhang, Simon Stepputtis, Katia P. Sycara, and Yaqi Xie. Dual prototype evolving for test-time generalization of vision-language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Marvin Mengxin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *Advances in Neural Information Processing Systems*, 2022.

Youjia Zhang, Youngeun Kim, Young-Geun Choi, Hongyeob Kim, Huiling Liu, and Sungeun Hong. Backpropagation-free test-time adaptation via probabilistic gaussian alignment. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jianchao Zhao, Chenhao Ding, SongLin Dong, Jiangyang Li, Qiang Wang, Yuhang He, and Yihong Gong. Shared & domain self-adaptive experts with frequency-aware discrimination for continual test-time adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(34): 28697–28705, 2026.

A Appendix

A.1 Experimental Setup

We use Point-MAE (Pang et al., 2022), DGCNN (Wang et al., 2019), and PointNet (Qi et al., 2017) as source models, each pretrained on the corresponding clean dataset. Unless otherwise specified, all experiments use a batch size of 32 and a learning rate of 0.001, with $\gamma = 0.98$, $\lambda = 10$, $t_0 = 0.4$, $t_1 = 0.8$, and $\alpha = 0.05$ across all model–dataset combinations. During adaptation, only the affine parameters of the normalization layers are updated, while the remaining network parameters are fixed. We perform one adaptation step and one prototype-refinement step for each incoming test batch. The prototype bank is initialized from the first target batch and subsequently updated online using soft class assignments. The reference topology $\mathbf{A}_{\mathrm{ref}}$ is computed once from the fixed source classifier weights and remains unchanged throughout adaptation.

Table 9: Performance on source (clean) data after adaptation to each target domain.

	Time		t ⟶														
	Model	clean	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean
MN-C	Point-MAE	89.22	89.75	89.47	89.55	88.74	88.82	89.55	89.10	89.47	89.02	89.06	88.65	88.82	89.55	89.25	88.33	89.14
	DGCNN	90.11	90.36	90.19	90.19	90.17	90.44	89.87	89.99	90.24	89.83	89.75	89.59	89.67	90.15	89.51	90.07	90.00
	PointNet	84.56	85.82	85.17	85.33	85.21	85.90	86.18	85.29	85.85	85.05	84.81	84.81	84.56	84.68	83.43	82.66	84.98
SONN-C	Point-MAE	76.94	77.80	77.45	75.73	76.76	76.94	77.97	77.28	77.28	77.28	77.11	76.94	77.11	77.11	77.45	76.59	77.12
	DGCNN	75.56	73.32	75.39	74.18	75.04	74.01	75.73	75.39	76.76	77.11	75.90	76.08	77.11	77.11	75.53	72.81	75.43
	PointNet	62.65	64.72	63.86	65.06	64.54	64.72	66.09	63.68	64.54	65.92	66.44	65.58	66.27	66.27	67.64	65.82	65.41
SN-C	Point-MAE	85.53	86.11	86.00	85.84	86..21	85.97	85.96	85.75	86.02	85.77	85.75	85.17	86.19	85.43	85.43	85.30	85.76
	DGCNN	85.85	85.98	85.84	85.63	85.60	86.04	85.71	85.65	85.40	85.62	85.47	85.55	85.61	85.67	85.76	85.39	85.66
	PointNet	82.34	81.96	81.66	82.40	81.72	81.34	80.73	81.21	81.24	81.78	81.58	82.33	82.25	82.03	82.28	82.16	81.78

Table 10: Mean accuracy (%) and standard deviation over 10 random domain orders.

Model	MN-C	SONN-C	SN-C
Point-MAE	75.05±0.2	50.34±0.2	68.54±0.2
DGCNN	77.01±0.4	50.43±0.2	66.79±0.3
PointNet	63.87±0.3	40.63±0.2	56.88±0.2

Computing Infrastructure. All experiments are conducted on a workstation running Ubuntu 22.04.4 LTS with two Intel Xeon Gold 5218R CPUs (20 cores / 40 threads each, 2.10 GHz) and 384 GB of system memory. Each experiment is run on a single NVIDIA RTX A6000 GPU with 48 GB of memory. Our implementation uses Python 3.8.16 and PyTorch 1.12.1 with CUDA 11.3 and cuDNN 8.3.2.

A.2 Preservation of Source Knowledge

To evaluate whether continual adaptation causes catastrophic forgetting of the source domain, we evaluate accuracy on the original clean data after adapting to each target corruption individually. As shown in Tab. 9, ToCoTTA largely preserves source-domain performance across all three datasets and backbones, with only marginal deviations from the original clean accuracy. In several cases, the source accuracy even slightly improves after adaptation, suggesting that the proposed updates do not substantially overwrite the pretrained source knowledge. These results indicate that ToCoTTA can adapt to corrupted target domains while effectively retaining the pretrained source knowledge.

A.3 Robustness to Domain Order

To evaluate sensitivity to the order of target domains, we randomly shuffle the corruption sequence using 10 different seeds and report the mean and standard deviation of the resulting accuracy. As shown in Tab. 10, ToCoTTA exhibits consistently low variance across all datasets and backbones. The small standard deviations indicate that its performance is largely insensitive to the specific domain order, suggesting robust continual adaptation under different corruption sequences.

A.4 Hyperparameter Sensitivity

We analyze the sensitivity of ToCoTTA to its main hyperparameters, including the learning rate, prototype bank decay γ, topology weight λ, restore rate α, and threshold pair (t_0, t_1). As shown in Fig. 5, ToCoTTA remains stable across a broad range of values for γ, λ, α, and (t_0, t_1), with only moderate sensitivity to the learning rate. Overall, these results demonstrate that ToCoTTA is robust to hyperparameter choices and does not require narrowly tuned settings.

A.5 Additional Quantitative Results

Tables 11–13 provide the full quantitative results on MN-C, SONN-C, and SN-C across all three backbones. Beyond the Point-MAE results reported in the main paper, ToCoTTA consistently achieves the highest mean accuracy with DGCNN and PointNet across all three datasets. On MN-C, ToCoTTA reaches 75.72% and 62.91% with DGCNN and PointNet, respectively, outperforming the corresponding strongest baselines. The same trend holds on SONN-C, where ToCoTTA achieves

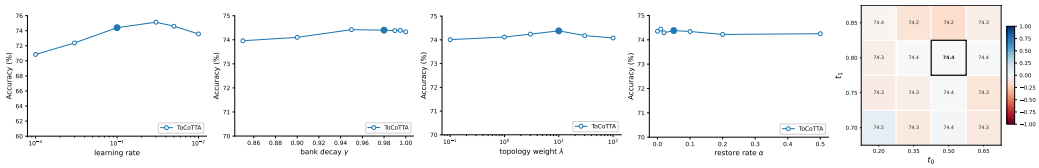


Figure 5: Hyperparameter sensitivity with respect to the learning rate, bank decay γ, topology weight λ, restore rate α, and thresholds (t_0, t_1).

Table 11: Classification accuracy (%) on ModelNet40-C across corruption types and backbones. Best and second-best results are highlighted in **bold** and underline, respectively.

	Time	t →															
	Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean
Point-MAE	Source-Only	66.57	59.16	7.01	31.73	65.56	67.71	69.81	77.96	85.25	74.35	38.09	74.76	70.02	38.53	30.47	57.13
	T3A	72.69	70.02	9.24	44.77	71.64	72.53	74.47	80.79	85.83	78.93	43.23	78.93	74.39	37.72	31.48	61.78
	DUA	79.62	77.88	17.22	68.27	63.09	72.73	76.30	84.08	85.39	78.85	52.67	80.63	76.58	41.82	40.80	66.39
	Purge-Gate†	78.12	74.47	**40.88**	66.69	81.73	73.58	75.81	79.09	82.37	77.23	58.19	75.49	73.58	47.85	45.14	68.68
	LN-TTA†	79.86	75.53	29.30	67.99	81.81	76.01	76.42	80.88	83.47	78.32	60.17	77.59	74.88	52.47	51.42	69.74
	PL	75.85	71.52	23.50	64.14	73.38	73.70	75.28	79.82	83.91	74.76	56.93	76.90	70.54	45.75	45.18	66.08
	TENT	75.73	72.08	22.93	64.87	73.66	73.78	75.73	80.23	83.71	75.53	58.27	77.27	70.46	46.56	45.87	66.45
	SHOT	75.53	75.85	29.35	70.30	74.47	75.89	74.11	77.43	78.28	67.38	54.38	71.43	65.36	46.03	43.23	65.27
	MEMO†	63.01	54.78	4.82	23.30	65.15	66.25	68.19	78.32	85.45	74.51	36.35	74.15	69.77	38.41	29.21	55.45
	SVWA	79.78	75.81	25.61	66.90	75.00	75.24	77.51	82.66	85.82	79.94	60.37	81.28	76.99	51.34	48.87	69.54
	CoTTA	80.21	78.66	29.15	69.83	78.15	78.38	78.84	83.68	85.28	81.69	66.92	81.93	78.55	53.97	55.16	72.03
	ViDA	79.82	76.99	25.53	67.91	76.34	77.07	77.51	83.02	86.18	80.51	63.29	81.85	78.36	51.26	51.22	70.46
	Hybrid-TTA	80.07	76.62	25.28	67.42	82.70	77.96	78.53	81.08	84.32	82.33	64.26	80.11	78.97	53.61	55.07	71.22
	ToCoTTA	**81.52**	**79.09**	30.62	**74.60**	**86.10**	**80.19**	**82.09**	**86.26**	**86.41**	**82.87**	**68.36**	**82.85**	**80.23**	**55.81**	**58.71**	**74.38**
DGCNN	Source-Only	68.48	75.04	12.72	54.98	64.30	74.47	75.24	74.23	84.64	76.78	57.46	74.23	76.38	32.37	24.43	61.72
	DUA	80.16	82.54	24.47	75.00	76.22	79.09	81.40	76.30	85.22	79.74	66.57	78.08	80.19	37.16	34.68	69.12
	PL	77.72	79.98	34.04	70.22	74.72	77.27	77.51	76.42	83.63	77.47	66.25	77.43	76.22	36.06	34.60	67.97
	TENT	77.39	80.59	35.41	70.87	75.57	78.00	78.61	77.11	84.52	78.65	68.56	77.88	77.59	39.22	36.35	69.09
	SHOT	77.88	80.35	35.86	73.42	75.53	79.21	78.69	76.54	83.47	78.73	69.00	77.19	76.86	35.66	32.25	68.71
	MEMO	64.87	70.87	10.29	48.34	60.13	71.56	73.99	73.50	84.32	75.49	54.94	72.85	74.31	31.12	24.60	59.41
	SVWA	81.08	82.82	41.01	75.20	79.54	81.56	82.13	79.51	85.06	80.96	73.02	82.78	79.78	44.04	39.91	72.56
	CoTTA	80.47	81.81	44.13	75.89	80.51	81.16	80.88	79.84	84.37	82.42	76.84	81.77	80.04	45.46	40.59	73.08
	ViDA	80.51	82.78	36.10	75.20	81.16	81.44	81.73	79.07	85.31	82.70	75.12	82.13	80.85	39.51	37.68	72.09
	ToCoTTA	**82.25**	**84.08**	**46.15**	**79.09**	**82.74**	**83.10**	**84.36**	**82.46**	**87.20**	**84.20**	**78.44**	**83.51**	**82.13**	**49.23**	**46.84**	**75.72**
PointNet	Source-Only	42.91	82.70	4.09	41.65	42.83	44.13	44.65	84.36	86.14	33.71	24.03	83.47	37.89	22.73	21.80	46.47
	DUA	70.58	82.58	6.32	64.38	70.34	60.78	64.75	81.24	84.81	42.10	30.07	81.65	51.99	34.97	37.84	57.63
	PL	70.46	82.54	13.05	64.83	70.42	61.06	64.06	82.98	83.39	46.68	36.26	80.67	56.20	30.31	32.50	58.36
	TENT	70.71	82.66	12.68	65.28	70.95	62.32	65.40	82.98	83.79	49.60	40.52	81.32	59.08	33.79	37.20	59.88
	SHOT	70.83	82.01	11.35	64.87	68.23	58.96	60.45	80.71	79.82	41.29	30.88	75.57	48.74	26.14	30.23	55.34
	MEMO	41.90	82.01	4.21	40.32	41.65	43.52	43.80	84.01	84.51	32.82	23.70	81.23	37.24	23.54	22.08	45.77
	SVWA	71.19	83.35	12.48	65.96	71.03	63.37	65.84	83.51	84.24	50.16	41.13	81.20	58.91	35.29	38.74	60.43
	CoTTA	70.56	82.62	12.88	67.26	71.68	64.59	66.37	83.55	84.48	51.70	41.01	81.85	58.59	36.47	40.33	60.93
	ViDA	70.02	82.17	11.75	64.18	70.02	61.95	64.22	82.82	84.12	45.46	33.91	81.20	53.12	28.20	29.09	57.48
	ToCoTTA	**71.27**	**83.83**	**13.53**	**69.57**	**74.35**	**67.30**	**69.33**	**84.64**	**84.93**	**54.13**	**45.95**	**83.06**	**63.25**	**37.20**	**41.29**	**62.91**

50.65% and 40.43%, and on SN-C, where it attains 66.37% and 56.88%. These results further confirm that the gains of ToCoTTA are consistent across different backbone architectures rather than being specific to a particular source model.

Long-Horizon Continual Adaptation. Tables 14–16 provide the full long-horizon results over 10 consecutive corruption rounds without model reset. Across all three datasets and backbones, ToCoTTA achieves the highest accuracy in every round and maintains stable performance throughout prolonged adaptation. On MN-C, ToCoTTA achieves mean accuracies of 75.65%, 77.12%, and 64.24% with Point-MAE, DGCNN, and PointNet, outperforming the strongest corresponding baselines by 2.41, 2.32, and 2.08 percentage points, respectively. Similar stability is observed on SONN-C, where ToCoTTA consistently maintains its advantage across all 10 rounds. The benefit is particularly pronounced on SN-C, where all baseline methods progressively degrade over repeated adaptation, whereas ToCoTTA remains nearly unchanged from the first to the final round across all three backbones. These results demonstrate that ToCoTTA effectively mitigates error accumulation and remains robust over long adaptation horizons.

Table 12: Classification accuracy (%) on ScanObjectNN-C across corruption types and backbones. Best and second-best results are highlighted in **bold** and underline, respectively.

	Time	t ⟶															
	Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean
Point-MAE	Source-Only	15.66	15.15	12.74	15.84	14.11	40.45	41.48	70.74	70.05	40.45	29.60	69.02	42.69	6.20	8.09	32.82
	T3A	17.56	18.76	15.32	18.59	15.66	44.41	48.37	73.32	73.15	41.31	32.19	71.60	46.13	6.37	9.12	35.46
	DUA	24.61	36.66	10.84	30.47	28.23	44.41	46.99	61.96	70.05	46.47	38.38	65.06	51.46	7.75	10.16	38.23
	Purge-Gate	40.96	54.73	21.86	36.49	39.59	55.25	58.35	75.56	74.01	56.80	48.71	75.56	58.69	7.06	9.47	47.54
	LN-TTA	43.20	57.49	**25.13**	40.28	43.37	56.63	60.59	76.25	74.01	56.28	50.95	77.63	60.59	7.40	9.81	49.31
	PL	30.98	44.58	12.91	32.19	31.67	50.09	53.70	71.77	69.88	50.78	45.96	74.01	54.56	7.40	8.95	42.63
	TENT	30.29	45.96	13.08	32.70	33.05	49.40	52.50	73.67	71.60	50.60	45.96	74.36	54.91	7.57	8.78	42.96
	SHOT	34.08	52.67	12.39	37.69	39.59	51.81	57.66	73.84	73.67	57.14	50.78	75.04	60.41	6.20	8.26	46.08
	MEMO	14.80	14.11	11.70	13.08	12.39	34.42	36.83	64.54	68.85	37.01	27.02	65.40	36.49	7.06	8.78	30.17
	SVWA	33.22	49.23	14.11	36.83	34.25	53.36	55.25	75.56	74.11	54.91	47.33	74.01	56.28	8.26	9.64	45.09
	CoTTA	37.87	51.64	13.08	36.66	35.97	54.39	56.45	73.32	72.98	57.66	49.40	74.18	57.14	7.92	9.47	45.87
	ViDA	33.91	45.78	13.43	36.83	33.39	53.36	53.87	75.22	72.98	54.91	46.82	74.18	56.45	8.26	10.12	44.63
	Hybrid-TTA	38.21	54.91	13.25	32.87	38.90	53.53	57.14	75.39	71.26	57.32	46.64	75.90	58.69	8.95	10.50	46.23
	ToCoTTA	**43.82**	**59.38**	23.00	**41.31**	**44.41**	**57.45**	**60.62**	**76.40**	**75.90**	**59.40**	**52.50**	**77.73**	**61.45**	**9.43**	**10.84**	**50.24**
DGCNN	Source-Only	21.52	28.92	25.47	20.65	15.49	36.49	36.15	65.06	63.17	37.18	31.84	63.34	35.11	8.95	9.02	33.22
	DUA	30.47	44.58	30.83	39.24	29.09	43.55	45.96	57.66	62.48	51.81	42.86	59.73	50.26	8.43	9.09	40.40
	PL	35.80	49.05	15.66	47.33	35.46	48.02	52.83	67.99	67.30	51.98	45.27	67.81	54.56	9.64	6.71	43.69
	TENT	35.97	49.40	16.01	47.68	36.49	47.68	53.70	68.16	67.81	52.15	46.13	67.99	54.56	9.12	6.71	43.97
	SHOT	36.15	48.54	16.87	50.09	34.94	49.40	53.36	69.54	68.68	53.70	48.88	69.02	53.53	9.12	7.23	44.60
	MEMO	18.93	25.47	19.45	16.70	13.43	32.36	33.56	60.93	57.49	33.56	29.26	58.18	32.01	9.64	8.95	29.99
	SVWA	37.18	49.91	18.24	49.74	36.32	53.36	56.45	74.18	71.26	55.25	51.64	71.95	55.25	9.29	6.71	46.45
	CoTTA	40.62	56.28	19.62	51.14	37.82	56.32	57.59	74.36	72.18	60.07	58.55	74.22	60.07	10.16	5.85	48.99
	ViDA	35.28	49.57	17.21	50.43	37.52	52.50	55.08	72.29	69.71	55.59	53.36	73.15	53.87	10.50	6.37	46.16
	ToCoTTA	**41.07**	**56.77**	**31.47**	**52.67**	**38.21**	**56.80**	**57.94**	**74.51**	72.05	**60.38**	**58.94**	**75.39**	**63.51**	**10.98**	**9.12**	**50.65**
PointNet	Source-Only	20.83	35.46	10.33	18.42	20.14	25.30	28.57	63.00	64.54	27.02	23.24	63.00	26.68	6.89	9.98	29.56
	DUA	34.77	52.67	10.67	35.11	38.55	30.47	35.11	59.04	59.04	34.94	29.95	62.31	37.01	7.06	9.64	35.75
	PL	39.24	59.38	14.46	33.91	40.96	35.80	38.55	65.75	63.51	36.15	33.22	61.79	38.04	8.61	9.33	38.58
	TENT	39.42	**60.24**	14.46	35.28	41.14	35.63	38.73	65.75	64.20	37.69	34.08	62.13	39.59	8.47	9.50	39.09
	SHOT	39.76	59.21	14.97	33.39	40.45	36.66	39.42	65.58	63.00	37.18	32.19	62.13	37.87	7.92	7.92	38.51
	MEMO	19.11	31.84	9.81	17.56	18.76	23.92	26.51	61.27	64.20	27.88	22.38	61.79	26.51	6.89	9.81	28.55
	SVWA	38.73	60.07	14.11	34.60	41.82	36.66	39.07	65.75	64.89	37.52	34.08	62.65	39.59	8.78	10.09	39.23
	CoTTA	41.24	60.12	14.58	36.27	42.64	35.38	39.16	66.84	64.57	40.16	34.27	64.26	39.76	9.04	10.67	39.93
	ViDA	41.31	58.52	15.15	33.91	40.79	36.15	36.15	64.03	63.17	34.77	31.50	63.00	37.18	9.02	9.81	38.30
	ToCoTTA	**41.57**	60.21	**15.46**	**36.50**	**43.17**	**37.87**	**40.45**	**67.13**	**65.40**	**40.76**	**34.42**	63.17	**41.14**	**9.12**	10.16	**40.43**

Table 13: Classification accuracy (%) on ShapeNet-C across corruption types and backbones. Best and second-best results are highlighted in **bold** and underline, respectively.

	Time	t →															
	Method	uni	gauss	backg	impul	upsam	rbf	rbf-inv	den-dec	dens-inc	shear	rot	cut	distort	oclsion	lidar	Mean
Point-MAE	Source-Only	71.93	66.29	9.91	57.80	64.62	73.86	74.69	84.49	83.71	78.70	53.33	83.54	75.38	11.82	9.69	59.98
	T3A	72.24	67.07	6.75	57.35	64.80	72.62	73.77	82.66	82.59	77.61	49.47	82.66	73.40	9.65	7.71	58.69
	DUA	76.00	72.58	12.01	60.11	65.18	76.73	77.66	83.39	83.35	80.02	62.04	83.76	76.75	10.97	9.51	62.00
	Purge-Gate	79.21	75.43	22.46	68.38	78.95	76.78	77.12	83.52	80.73	79.24	64.91	83.55	76.08	10.15	9.55	64.40
	LN-TTA	79.18	76.62	14.65	68.72	79.28	77.95	78.20	84.38	82.32	79.43	65.25	83.91	77.12	9.11	9.41	64.37
	PL	72.20	68.11	8.53	65.45	68.20	74.31	74.65	82.42	81.76	76.84	61.29	81.44	75.08	9.71	9.95	60.66
	TENT	72.23	68.31	8.69	65.69	68.38	74.55	75.07	82.52	81.94	77.20	62.11	81.56	75.55	10.32	10.50	60.97
	SHOT	67.75	51.38	12.59	36.39	32.93	30.49	29.68	32.11	29.26	23.13	19.99	26.24	21.46	4.74	4.56	28.18
	MEMO	73.79	68.24	6.88	57.78	66.24	74.17	75.39	83.26	82.84	79.39	54.42	83.31	76.05	11.34	8.82	60.13
	SVWA	75.68	72.21	9.20	69.31	72.07	77.38	78.04	83.94	83.65	79.85	66.71	83.42	77.83	10.30	10.78	63.36
	CoTTA	77.28	76.46	13.71	72.46	74.59	78.44	78.91	82.45	82.19	80.89	73.41	81.64	79.24	10.99	11.99	64.98
	ViDA	75.36	73.01	9.83	70.32	73.40	77.87	78.53	83.59	83.20	80.46	68.77	83.29	78.05	9.97	10.08	63.71
	Hybrid-TTA	80.61	78.06	15.06	71.99	81.29	79.57	79.92	84.39	81.27	81.04	71.65	84.12	79.69	11.79	11.35	66.12
	ToCoTTA	**81.19**	**80.28**	**38.17**	**73.09**	**81.50**	**80.30**	**81.15**	**85.41**	**83.80**	**81.82**	**73.57**	**84.26**	**79.87**	**11.90**	**12.08**	**68.56**
DGCNN	Source-Only	69.26	67.93	26.40	60.69	62.15	74.92	75.89	82.96	82.37	75.99	63.02	82.86	74.53	7.85	7.11	60.93
	DUA	73.44	73.31	25.50	42.05	65.96	75.57	77.50	83.35	81.70	77.77	68.84	82.85	76.13	8.90	10.57	61.56
	PL	72.20	68.11	8.53	65.45	68.20	74.31	74.65	82.42	81.76	76.84	61.29	81.44	75.08	9.71	9.95	60.66
	TENT	70.18	70.79	13.24	69.85	65.24	75.04	75.39	81.59	81.34	75.53	66.56	81.10	74.11	8.98	11.55	61.37
	SHOT	65.72	60.80	9.56	53.79	45.67	51.70	49.35	54.72	50.84	44.33	39.13	48.35	39.88	3.31	2.87	41.33
	MEMO	68.56	66.70	22.88	53.49	61.09	74.34	74.94	82.68	82.25	75.61	64.45	82.52	73.98	7.39	6.25	59.81
	SVWA	73.06	73.14	13.99	70.63	67.75	77.18	77.61	82.61	82.50	78.40	70.17	82.34	76.13	8.73	10.94	63.01
	CoTTA	77.28	76.46	13.71	70.46	74.59	78.44	78.91	82.45	82.19	80.89	74.41	81.64	79.24	9.99	11.99	64.84
	ViDA	73.55	74.41	14.00	70.21	71.79	78.25	78.71	83.29	**82.90**	79.75	74.37	83.05	78.62	9.47	11.82	64.28
	ToCoTTA	**78.04**	**77.68**	**26.64**	**70.65**	**75.14**	**78.50**	**80.42**	**84.50**	82.66	**80.91**	**75.22**	**83.86**	**79.74**	**10.18**	11.47	**66.37**
PointNet	Source-Only	59.29	75.03	9.06	52.71	58.94	55.63	55.39	82.15	81.63	43.10	26.26	80.89	48.53	6.42	5.49	49.37
	DUA	67.12	77.37	12.21	39.84	64.57	58.89	61.17	82.54	81.67	45.85	34.17	80.80	52.37	8.81	9.52	51.79
	PL	64.61	77.61	12.18	55.92	65.88	58.29	60.12	82.19	81.67	45.78	34.28	80.98	52.38	9.12	9.02	52.67
	TENT	64.68	77.61	12.30	56.44	66.24	59.01	60.81	82.15	81.77	46.91	35.72	81.27	54.01	9.32	9.26	53.17
	SHOT	61.86	69.11	5.61	33.43	34.01	28.21	29.37	43.86	38.31	16.89	12.46	34.51	18.65	3.32	3.00	28.84
	MEMO	58.95	74.57	9.28	53.35	59.00	56.11	55.85	82.39	81.75	43.47	26.63	80.84	48.90	6.67	5.59	49.56
	SVWA	64.59	77.95	11.65	56.34	66.09	59.04	60.88	82.15	81.40	46.95	35.50	81.60	53.82	9.01	8.79	53.05
	CoTTA	65.72	77.48	15.13	60.39	67.46	63.62	64.58	81.52	81.06	54.96	**42.97**	81.39	59.95	12.23	10.75	55.95
	ViDA	64.79	77.61	8.10	55.33	64.82	57.07	57.18	80.75	79.89	35.29	24.07	75.94	34.21	3.86	3.28	48.14
	ToCoTTA	**68.74**	**78.39**	**15.54**	**61.36**	**69.84**	**65.42**	**66.44**	**82.99**	**81.80**	**55.56**	42.92	**81.94**	**59.98**	11.49	**10.80**	**56.88**

Table 14: Long-horizon adaptation on ModelNet-C over 10 corruption rounds without model reset.

	Round	R →										
	Method	1	2	3	4	5	6	7	8	9	10	Mean
Point-MAE	CoTTA	72.03	72.93	72.81	73.39	73.20	73.23	73.67	73.27	73.79	74.08	73.24
	ViDA	70.46	71.51	71.77	72.39	72.78	73.00	73.32	73.09	73.32	73.89	72.55
	Hybrid-TTA	71.22	71.55	72.55	73.16	73.14	73.07	73.42	73.61	73.57	73.47	72.88
	ToCoTTA	**74.38**	**75.00**	**75.33**	**75.56**	**75.59**	**76.03**	**76.31**	**76.16**	**75.83**	**76.34**	**75.65**
DGCNN	CoTTA	73.08	73.55	74.54	74.85	75.16	75.51	75.46	75.37	75.38	75.15	74.80
	ViDA	72.09	72.50	73.07	73.49	74.12	74.37	74.18	74.38	74.45	74.38	73.70
	ToCoTTA	**75.72**	**76.90**	**77.23**	**77.53**	**77.32**	**77.32**	**77.38**	**77.15**	**77.15**	**77.52**	**77.12**
PointNet	CoTTA	60.93	61.13	61.72	62.28	62.41	62.59	62.50	62.66	62.71	62.70	62.16
	ViDA	57.48	57.71	58.26	57.47	57.25	57.12	55.72	54.57	52.71	50.19	55.85
	ToCoTTA	**62.91**	**64.40**	**64.35**	**64.18**	**64.28**	**64.13**	**64.31**	**64.68**	**64.40**	**64.75**	**64.24**

Table 15: Long-horizon adaptation on ScanObjectNN-C over 10 corruption rounds without model reset.

	Round	R →										
	Method	1	2	3	4	5	6	7	8	9	10	Mean
Point-MAE	CoTTA	45.87	45.96	45.94	46.10	46.57	47.34	47.37	47.26	47.37	47.56	46.73
	ViDA	44.63	45.06	45.33	46.14	46.45	46.07	45.85	46.13	46.33	46.38	45.84
	Hybrid-TTA	46.23	45.19	46.89	46.13	46.21	46.30	46.77	46.54	46.85	47.01	46.41
	ToCoTTA	**50.24**	**50.73**	**50.58**	**51.31**	**50.73**	**51.09**	**51.23**	**51.35**	**51.18**	**51.48**	**50.99**
DGCNN	CoTTA	48.99	48.94	48.50	48.98	49.21	49.15	49.04	49.67	49.81	49.85	49.21
	ViDA	46.16	46.55	46.93	46.96	47.23	47.34	47.34	47.24	47.15	47.51	47.04
	ToCoTTA	**50.65**	**50.92**	**50.72**	**50.61**	**50.64**	**50.72**	**50.42**	**50.98**	**50.93**	**50.95**	**50.75**
PointNet	CoTTA	39.93	39.50	40.00	39.71	40.02	40.13	40.60	40.82	41.08	40.76	40.25
	ViDA	38.30	39.04	38.61	38.11	38.81	38.69	38.45	38.84	38.24	38.20	38.53
	ToCoTTA	**40.43**	**41.70**	**41.82**	**41.00**	**41.34**	**41.04**	**40.68**	**41.34**	**40.75**	**41.21**	**41.13**

Table 16: ong-horizon adaptation on ShapeNet-C over 10 corruption rounds without model reset.

	Round	R →										
	Method	1	2	3	4	5	6	7	8	9	10	Mean
Point-MAE	CoTTA	64.98	64.46	64.07	64.18	64.07	63.99	63.70	64.00	63.79	63.95	64.12
	ViDA	63.71	64.54	64.27	63.61	62.88	60.84	59.30	58.81	57.22	55.84	61.10
	Hybrid-TTA	66.12	65.93	66.04	66.09	65.73	65.38	65.29	65.41	65.27	65.31	65.66
	ToCoTTA	**68.56**	**68.71**	**68.66**	**68.73**	**68.69**	**68.70**	**68.56**	**68.48**	**68.66**	**68.68**	**68.64**
DGCNN	CoTTA	64.84	63.48	60.05	56.11	52.21	50.04	48.68	47.39	46.88	46.44	53.61
	ViDA	64.28	64.63	62.74	61.20	61.20	59.04	57.44	55.51	53.70	51.97	59.17
	ToCoTTA	**66.37**	**66.36**	**66.35**	**66.26**	**66.34**	**66.23**	**65.91**	**66.36**	**66.24**	**66.09**	**66.25**
PointNet	CoTTA	55.95	56.18	55.47	55.79	54.49	53.12	53.05	52.54	51.17	50.17	53.79
	ViDA	48.14	9.36	2.79	1.93	2.10	2.16	1.94	1.89	1.77	1.72	7.38
	ToCoTTA	**56.88**	**56.65**	**56.51**	**56.50**	**56.75**	**56.38**	**56.50**	**56.43**	**56.54**	**56.34**	**56.55**