GoTTA be Diverse: Rethinking Memory Policies for Test-Time Adaptation

Anonymous authors
Paper under double-blind review

Abstract

Test-time adaptation (TTA) adapts pretrained models online to unlabeled test streams under distribution shift. Under temporally correlated and label-skewed streams, memory buffers determine which samples are repeatedly used for adaptation, yet memory selection is often coupled with a specific TTA method, making its contribution difficult to isolate. We decouple memory selection from the adaptation objective and benchmark six memory policies across eleven TTA methods under temporally correlated streams, covering both episodic and continual adaptation. Our analysis shows that memory selection can substantially affect TTA performance and that its impact depends on the underlying adaptation mechanism: pseudo-labeling methods can be sensitive to redundant samples, while normalization- and RoTTA-based methods exhibit different responses to diversity-aware retention. We identify within-class redundancy as a complementary limitation of class-aware memory selection. Although class-balanced policies regulate how memory capacity is distributed across pseudo-classes, they do not explicitly prevent multiple slots from containing highly similar samples. Motivated by this observation, we introduce Guided Observational Test-Time Adaptation (GOTTA), a memory policy that combines class-aware allocation with within-class diversity filtering. GOTTA measures diversity in softmax predictive-distribution space while retaining the existing retention-utility mechanism and leaving the underlying adaptation objective unchanged. Across our experiments, diversity-aware memory selection improves a broad range of TTA methods under temporally correlated streams, with the magnitude of the benefit depending on the adaptation method and available memory capacity. Experiments on CIFAR-10-C demonstrate the importance of memory composition under constrained budgets, while evaluations on ImageNet-C and ITD examine its behavior across datasets and stream settings. Overall, our study identifies memory selection as a distinct design axis of TTA and shows that how samples are retained can substantially shape online adaptation. These findings motivate treating memory policy as a first-class design choice alongside the adaptation objective.

1 INTRODUCTION

Deep neural networks are typically trained under the assumption that training and test samples are independently and identically distributed (i.i.d.). In deployment, however, models encounter test distributions that shift over time due to changes in appearance, environment, acquisition conditions, or class prevalence (Quinonero-Candela et al., 2008). **Test-Time Adaptation (TTA)** addresses this problem by adapting a pre-trained, source-free model online using only the incoming unlabeled test stream, without revisiting source data (Wang & Deng, 2018; Zhou et al., 2022). Early TTA methods address static or mildly varying target distributions through batch normalization recalibration (Schneider et al., 2020; Nado et al., 2020), entropy minimization (Wang et al., 2021), pseudo-labeling (Lee et al., 2013), or self-supervised objectives (Sun et al., 2020). However, practical deployment often involves **dynamic, non-i.i.d. streams**, where gradual appearance changes and abrupt class-prevalence shifts produce temporally correlated, imbalanced, and visually redundant samples. Such streams can amplify prediction errors and accelerate forgetting.

Recent continual TTA methods therefore employ memory replay buffers to stabilize adaptation (Gong et al., 2022; Yuan et al., 2023). These methods typically select samples using criteria

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.

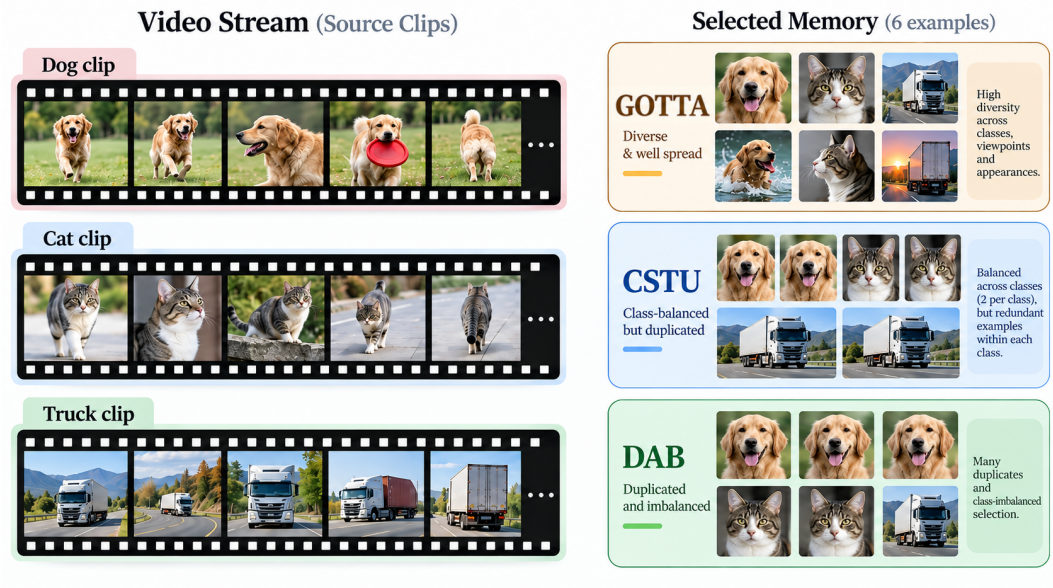

Figure 1: Illustration of memory selection under a streaming video setting. The incoming stream contains temporally correlated clips from different classes. **GOTTA (OURS)** promotes diversity by retaining visually distinct samples, **CSTU** maintains class-balanced memory but may preserve redundant samples within each class, while **DAB** can retain duplicated and class-imbalanced samples.

such as class balance, confidence, uncertainty, or recency. However, memory policies are often evaluated only as components of specific TTA algorithms, making it difficult to isolate which memory design choices are beneficial across adaptation objectives and stream conditions.

In this work, we decouple memory selection from the adaptation algorithm and systematically study how memory composition affects test-time adaptation. Our analysis shows that memory selection can substantially affect TTA performance, and that its impact depends on the underlying adaptation mechanism. In particular, class-aware memory policies can control how memory capacity is distributed across pseudo-classes, but they do not explicitly control redundancy among samples assigned to the same class. While DAB (Döbler et al., 2024) introduces diversity-aware buffering by favoring samples whose predictions differ from the model's recent output tendency, its diversity criterion operates at the level of the model's recent predictions rather than explicitly measuring redundancy within each pseudo-class. Under a limited memory budget, multiple retained samples can therefore still have highly similar predictive distributions even when the memory is well balanced across pseudo-classes. This issue is particularly relevant in temporally correlated streams, where consecutive observations may provide highly overlapping adaptation signals.

This motivates a complementary question: *given that memory capacity is allocated to a pseudo-class, which samples should be retained so that the memory covers diverse predictive states rather than repeatedly storing similar observations?* We view this question as distinct from the design of the underlying TTA objective. The adaptation method determines how stored samples are used for updating the model, whereas the memory policy determines which observations are available to that method. This separation allows us to study memory composition as an independent component of TTA.

We therefore propose **Guided Observational Test-Time Adaptation (GOTTA)**, a diversity-aware memory module that explicitly promotes diversity within each pseudo-label class. GOTTA is a drop-in replacement for existing replay buffers: it combines class-aware allocation with within-class diversity filtering in softmax predictive-distribution space. In addition, it retains the existing retention utility based on sample age and predictive uncertainty, allowing us to isolate the contribution of the proposed diversity criterion. Rather than modifying the adaptation objective or update rule, GOTTA changes which samples are retained in memory, with the goal of reducing within-class redundancy under constrained memory budgets.

Figure 1 illustrates this principle in a temporally correlated video stream: class-guided memory can maintain balanced class occupancy while retaining redundant samples, whereas GOTTA explicitly promotes diversity within each pseudo-class while preserving class coverage. Across multiple TTA objectives, memory budgets, and non-i.i.d. stream settings, our experiments show that diversity-aware memory selection can improve adaptation across a broad range of settings, while the magnitude of the benefit depends on the underlying adaptation mechanism and memory capacity.

Contributions.

- We decouple memory selection from the adaptation algorithm and provide a systematic evaluation of memory policies across multiple TTA methods and non-i.i.d. stream settings, showing that memory composition can substantially affect adaptation performance.
- We identify **within-class redundancy** as a complementary limitation of class-aware memory selection: balancing pseudo-class occupancy does not prevent multiple memory slots from containing highly similar predictive states.
- We propose Guided Observational Test-Time Adaptation (GOTTA), a diversity-aware memory module that combines class-aware allocation with within-class diversity filtering in softmax predictive-distribution space, while leaving the underlying TTA objective unchanged.

2 RELATED WORKS

Unsupervised domain adaptation. Unsupervised domain adaptation (UDA) adapts models from labeled source data to unlabeled target data under domain shift (Li et al., 2022; Wang & Deng, 2018). Common approaches align source and target distributions using statistical distances (Long et al., 2019), adversarial domain confusion (Ganin et al., 2016; Saito et al., 2018), or pseudo-label-based self-training (Xie et al., 2023), often with class balancing or confidence filtering to reduce pseudo-label bias (Zou et al., 2018). Unlike UDA, TTA operates online without access to source data and adapts solely from the incoming unlabeled test stream.

Continual learning. Continual learning (CL) addresses sequential learning while mitigating catastrophic forgetting (Lange et al., 2022). Existing approaches rely on parameter regularization (Kirkpatrick et al., 2016; Aljundi et al., 2019), parameter isolation, or rehearsal of stored examples (Castro et al., 2018; Rebuffi et al., 2017). Our work is related to rehearsal-based CL through its use of a compact memory buffer, but differs in operating under continuous, unlabeled, source-free test streams with unknown and shifting distributions.

Test-time adaptation and practical streams. TTA adapts pretrained models during inference using only unlabeled test samples (Liang et al., 2025). Early methods considered static or approximately i.i.d. shifts through batch-normalization statistics (Ioffe & Szegedy, 2015; Nado et al., 2020; Schneider et al., 2020), entropy minimization (Wang et al., 2021), pseudo-labeling (Lee et al., 2013), and self-supervised objectives (Sun et al., 2020), with subsequent methods adapting selected or full model parameters (Liang et al., 2020; Jang & Chung, 2022; Iwasawa & Matsuo, 2021). More recent work considers continual and practical TTA, where temporal correlation, distribution drift, redundancy, and class imbalance become important (Wang et al., 2022; Gong et al., 2022; Yuan et al., 2023). NOTE (Gong et al., 2022) and RoTTA (Yuan et al., 2023) incorporate memory for temporally correlated streams, while ITD (Alhuwaider et al., 2025) provides a tracklet-based video benchmark with inherently non-i.i.d. structure. We follow this practical TTA setting but focus on memory selection rather than introducing a new adaptation objective.

Memory mechanisms in practical TTA. Memory buffers stabilize adaptation under temporally correlated streams (Gong et al., 2022; Yuan et al., 2023; Yu et al., 2024). Existing policies range from uninformed FIFO and reservoir sampling (Vitter, 1985), to pseudo-label-based class balancing (Gong et al., 2022; Yuan et al., 2023) and selective filtering of unreliable or outlier samples (Yu et al., 2024). However, these policies are commonly coupled to particular TTA objectives, making their independent contribution difficult to assess. We instead decouple memory selection from adaptation and evaluate memory policies under a unified protocol across methods and streams.

Geometric diversity and coreset selection. Temporally correlated streams can contain substantial redundancy, motivating memory selection as online coreset construction. (Sener & Savarese, 2017; Giancola et al., 2023) study diversity-based selection in active learning and coreset construction.

Representative approaches include farthest point sampling (Eldar et al., 1997; Qi et al., 2017) and determinantal point processes (Kulesza & Taskar, 2012; Chen et al., 2018), while DAB (Döbler et al., 2024) applies diversity-aware replay to TTA. DAB measures diversity relative to a recent prediction trend using an exponential moving average, whereas our approach measures pairwise diversity among samples within each pseudo-label partition. This distinction enables the memory to maintain class coverage while reducing within-class redundancy. Unlike offline coreset selection, our setting is online, source-free, pseudo-labeled, and continuously affected by the evolving predictions of the adapting model.

3 METHODOLOGY

3.1 PROBLEM DEFINITION

Let $f_{\theta_s} : \mathcal{X} \to \mathcal{Y}$ denote a model pre-trained on a labeled source dataset $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ drawn from a source distribution $\mathcal{P}_s$. At deployment, test-time adaptation (TTA) assumes access only to the pre-trained parameters θ_s; the source data $\mathcal{D}_s$ is unavailable.

The model receives a continuous stream of unlabeled test batches $\{\mathcal{X}_t\}_{t=1}^{T}$, where samples at time t are drawn from a target distribution $\mathcal{P}_t$ that may differ from $\mathcal{P}_s$. Unlike conventional domain adaptation settings with a fixed target distribution, practical test streams may be non-stationary, such that $\mathcal{P}_t \neq \mathcal{P}_{t-1}$, due to changes in appearance, temporal correlations, or class prevalence. At each time step t, a TTA method updates the model using an unsupervised adaptation objective $\mathcal{L}_{\text{adapt}}$:

$$\theta_t \leftarrow \text{Adapt}\big(\theta_{t-1}, \mathcal{X}_t; \mathcal{L}_{\text{adapt}}\big).$$

Consequently, the model update depends not only on the adaptation objective, but also on the empirical distribution of samples used to perform that update. In non-i.i.d. streams, adapting directly to the current batch can expose the model to temporally redundant or class-imbalanced samples, potentially biasing successive updates and amplifying errors over time.

3.2 TEST-TIME ADAPTATION WITH MEMORY

Memory-augmented TTA maintains a fixed-capacity buffer $\mathcal{M}$ of previously observed test samples and uses the retained samples to support subsequent adaptation. At time t, the memory contains at most M entries,

$$\mathcal{M}_t = \{m_1, \ldots, m_K\}, \qquad K \leq M.$$

Each memory entry consists of an observed sample and model-derived metadata, $m_i = (x_i, \hat{y}_i, u_i, z_i, a_i)$, where x_i is the input sample, $\hat{y}_i$ is its pseudo-label, u_i is its predictive uncertainty, z_i is its predictive distribution, and a_i denotes its age in memory. Details on memory organization across class partitions are provided in Appendix A.1. Given a C-class model, we define the predictive distribution as

$$z_i = \text{softmax}(f_\theta(x_i)) \in \Delta^{C-1},$$

where Δ^{C-1} denotes the probability simplex over the C classes. The corresponding pseudo-label and predictive uncertainty are

$$\hat{y}_i = \arg\max_c z_{i,c}, \qquad u_i = -\sum_{c=1}^{C} z_{i,c} \log z_{i,c}.$$

At each time step, the memory is updated according to a memory policy Π,

$$\mathcal{M}_t \leftarrow \Pi(\mathcal{M}_{t-1}, \mathcal{X}_t),$$

after which the retained samples are used for adaptation,

$$\theta_t \leftarrow \text{Adapt}\big(\theta_{t-1}, \mathcal{M}_t; \mathcal{L}_{\text{adapt}}\big).$$

The policy Π determines which observed samples are retained or discarded under the memory budget M. Importantly, Π does not modify the underlying adaptation objective. Instead, it controls the empirical distribution of samples presented to that objective.

3.3 A TAXONOMY OF MEMORY POLICIES

Memory policies differ in the information they use to determine which samples should be retained under a fixed memory budget.

Uninformed memories. Uninformed policies make retention decisions without using model predictions or predictive representations. FIFO prioritizes recency by evicting the oldest samples, whereas reservoir sampling maintains a uniform random sample of the observed stream (Vitter, 1985). Because these policies do not explicitly account for the structure of the incoming stream, their memories may reflect temporal redundancy or class imbalance.

Class-guided memories. Class-guided policies use model predictions to regulate memory composition. PBRS applies reservoir sampling within pseudo-label partitions to reduce domination by frequently predicted classes. CSTU additionally incorporates timeliness and predictive uncertainty when determining which samples to retain (Yuan et al., 2023). However, class-level allocation alone does not explicitly control redundancy among samples assigned to the same pseudo-class.

Diversity-aware memories. Diversity-aware policies aim to reduce redundancy while retaining informative samples. DAB (Döbler et al., 2024) favors samples whose predictions deviate from the recent prediction tendency, while Guided Observational Test-Time Adaptation (GOTTA) explicitly promotes diversity within each pseudo-class. Specifically, GOTTA selects samples that expand the coverage of the stored predictive distributions, treating the memory as a compact online coreset of the test stream. For each incoming sample, GOTTA measures its distance to the nearest retained neighbor within the corresponding pseudo-class, thereby directly controlling within-class redundancy.

We introduce Guided Observational Test-Time Adaptation (GOTTA), which combines class-aware memory allocation with within-class geometric diversity. We view the resulting memory as an online coreset: rather than retaining samples solely according to class membership, recency, or uncertainty, GOTTA seeks to reduce redundancy among the predictive distributions retained within each pseudo-class.

3.4 GUIDED OBSERVATIONAL TEST-TIME ADAPTATION: DIVERSITY-AWARE MEMORY

GOTTA is built on the principle that class coverage and within-class diversity should be considered jointly when constructing memory. Each incoming sample is assigned to a pseudo-label partition, while insertion and eviction decisions account for predictive uncertainty, sample age, and geometric redundancy. The class-wise allocation limits domination by frequent pseudo-classes, while the diversity criterion reduces redundancy within each partition. We measure diversity in the softmax predictive space because these distributions directly characterize the model's prediction for each sample. Similar predictive distributions may provide redundant adaptation signals, whereas differences indicate distinct prediction behavior. Moreover, the predictive distribution is compact, with dimensionality equal to the number of classes, making it efficient to store and compare for online memory selection.

Farthest Point Sampling. GOTTA promotes within-class diversity by comparing each incoming sample with the samples already retained in its pseudo-label partition. We refer to this mechanism as *Farthest Point Sampling* (FPS) for brevity. Unlike the standard offline FPS procedure, our implementation is an online nearest-neighbor diversity filter coupled with class-aware eviction.

Let $m_t = (x_t, \hat{y}_t, u_t, z_t, a_t)$ denote an incoming entry, and let $\mathcal{M}^{(\hat{y}_t)}$ denote the memory partition associated with its pseudo-label. The candidate is first evaluated by the within-class diversity filter. If $\mathcal{M}^{(\hat{y}_t)}$ is empty, the candidate is accepted automatically. Otherwise, we compute its distance to the nearest retained sample in the same pseudo-class:

$$d_{\min}(m_t) = \min_{m_i \in \mathcal{M}^{(\hat{y}_t)}} \|z_t - z_i\|_2.$$

The candidate passes the diversity filter only if $d_{\min}(m_t) > \epsilon$. Thus, the first sample assigned to a pseudo-class is always admitted, whereas subsequent samples must be sufficiently different from the

samples already retained in that partition. The threshold ϵ controls the minimum separation between predictive distributions within a pseudo-class. We study its effect in Section 4.3.

Once a candidate passes the diversity filter, it is inserted directly if free memory is available and the corresponding pseudo-class partition has not reached its capacity. If the global memory is full, the candidate must replace an existing entry. To preserve class balance, eviction is restricted to samples from the current majority pseudo-class(es). If the incoming pseudo-class has already reached its capacity, replacement is instead restricted to that same pseudo-class. Since pseudo-labels may be incorrect, we evaluate the robustness of this class-aware allocation using a dynamic memory-refresh variant, GOTTA-D, in Appendix A.6.

Within the permitted replacement set $\mathcal{R}_t$, we identify the stored sample closest to the incoming candidate in predictive distribution space:

$$m_r = \arg\min_{m_i \in \mathcal{R}_t} \|z_t - z_i\|_2.$$

The candidate replaces m_r only when the stored entry is no more desirable to retain than the incoming one, as determined by the shared eviction score $\mathcal{H}$: $\mathcal{H}(m_r) \geq \mathcal{H}(m_t)$. Otherwise, the candidate is discarded. When multiple majority pseudo-classes are eligible for eviction, we select the closest eligible neighbor across the corresponding partitions. To determine retention desirability, GOTTA uses the same CSTU-style heuristic score (Yuan et al., 2023) used by the corresponding memory baselines:

$$\mathcal{H}(m_i) = \lambda_t \frac{1}{1 + \exp(-a_i/M)} + \lambda_u \frac{u_i}{\log C},$$

where a_i is the age of the entry, u_i is its predictive uncertainty, C is the number of classes, and M is the memory capacity. The coefficients λ_t and λ_u control the relative contributions of age and uncertainty. Larger values of $\mathcal{H}(m_i)$ indicate entries that are less desirable to retain, making older or more uncertain samples more likely to be evicted. This score is shared with the competing memory policies; the key addition in GOTTA is the within-class diversity criterion, which complements class-aware memory allocation by reducing redundancy among samples assigned to the same pseudo-class. We ablate the contribution of this shared heuristic in Appendix A.7.

Compatibility and modularity. GOTTA modifies only the memory policy, leaving the adaptation objective, model architecture, and optimization procedure unchanged. It can therefore be integrated with different TTA methods without modifying their underlying adaptation mechanisms.

4 Experiments

Datasets and streams. We evaluate on CIFAR-10-C, ImageNet-C (Hendrycks & Dietterich, 2019), and ITD (Alhuwaider et al., 2025). CIFAR-10-C and ImageNet-C contain 15 common corruption types at five severity levels; following standard TTA protocols (Wang et al., 2021), we evaluate at severity level 5. For computationally tractable evaluation across the full set of memory policies and adaptation methods, the main paper evaluates a fixed 10-class subset of ImageNet-C, denoted as **ImageNet-C***. Results on the full 1,000-class ImageNet-C benchmark are reported in Appendix A.8.We additionally evaluate on ITD to study adaptation under temporally evolving video streams, with further setup details provided in Appendix A.2.

We evaluate all methods under temporally correlated non-i.i.d. streams following the Practical Test-Time Adaptation (PTTA) protocol (Gong et al., 2022; Yuan et al., 2023), where class proportions are controlled by a Dirichlet distribution γ. We follow the same class ordering and stream construction across all methods to ensure a consistent comparison. The exact class order and stream-generation details are provided in the Appendix A.3.

Adaptation protocol and metrics. We evaluate both episodic and continual adaptation. In the episodic setting, the model is reset to its source parameters at the beginning of each corruption, whereas in the continual setting, adaptation proceeds across the corruption sequence without resetting the model. We report top-1 classification accuracy, averaged across test streams and corruption types when multiple corruptions are evaluated. Table 1 and 4 report averages over three random seeds (42, 3407, and 12345), while all other tables use seed 42. For multi-seed experiments, the same stream realizations are used across memory policies for controlled comparison.

Table 1: Memory-constrained comparison on **CIFAR-10-C** in continual temporal streams with $M = 32$ for all memory-based methods with Dirichlet concentration settings $\gamma = 10^{-1}$. We report the mean accuracy averaged over three random seeds $\{42, 3407, 12345\}$; the corresponding results with standard deviations are provided in the supplementary material. We mark — for the no-memory baseline. the best result overall is highlighted in blue

Policy	Memory	Source	Energy	SHOT	TENT	SAR	EATA	CoTTA	PL	NOTE	Norm	RoTTA	DAB
Uninformed	—	**56.49**	12.29	16.48	15.55	**24.93**	**24.80**	22.50	13.78	22.88	24.80	50.21	25.01
	FIFO	–	10.89	12.44	12.20	23.15	22.98	20.87	11.33	43.17	67.67	64.16	68.41
	Reservoir	–	10.89	13.10	12.29	23.19	23.00	21.84	10.45	52.30	50.49	40.89	49.08
Class-guided	PBRS	–	11.32	13.76	14.98	22.86	23.00	21.58	15.10	69.97	72.60	68.89	71.48
	CSTU	–	10.96	15.91	15.11	22.99	22.99	22.13	13.99	69.91	71.41	69.62	71.24
Diversity-aware	DAB-buffer	–	12.06	15.03	17.37	22.94	22.97	22.35	18.93	**70.95**	59.49	57.9	58.30
	GOTTA (OURS)	–	**22.23**	**22.25**	**22.45**	22.99	22.99	**22.99**	**41.55**	68.72	**75.41**	**75.66**	**76.06**

TTA baselines and memory modules. We evaluate memory policies across representative TTA methods, including TENT (Wang et al., 2021), SHOT (Liang et al., 2020), PL (Lee et al., 2013), CoTTA (Wang et al., 2022), NOTE (Gong et al., 2022), RoTTA (Yuan et al., 2023), EATA (Niu et al., 2022), SAR (Niu et al., 2023), and Energy (Yuan et al., 2024). We additionally report the **Source** model, which evaluates the pre-trained model without test-time adaptation. Unless otherwise specified, each TTA method follows the optimization configuration of its original implementation. We compare memory policies according to the taxonomy in Section 3.3. We additionally evaluate each adaptation method without memory to isolate the effect of memory selection.

Models. We adopt standard backbones for each benchmark: WRN-28-10-BN (Zagoruyko & Komodakis, 2016) for CIFAR-10-C, ResNet-50 (He et al., 2016) for ImageNet-C, and ResNet-18 for ITD, where the backbone is pretrained on the ITD training split following Alhuwaider et al. (2025). All experiments are conducted on a single NVIDIA V100 or A100 GPU. Additional results with ViT Dosovitskiy et al. (2021) on ImageNet-C are provided in Appendix A.9.

4.1 Comparison with Memory Policies

Table 1 compares memory policies under a constrained budget of $M = 32$ on continual, temporally correlated CIFAR-10-C streams. Under this identical capacity constraint, GOTTA achieves the best or tied-best result for most evaluated adaptation methods, while the magnitude of the benefit varies substantially across objectives. Improvements are most pronounced for Norm (75.41%) and RoTTA (75.66%), outperforming CSTU (71.41% and 69.62%) and DAB-buffer (59.49% and 57.90%). GOTTA also substantially boosts PL to 41.55%, compared to 18.93% for DAB-buffer and 15.10% for PBRS.

These gains highlight the importance of mitigating sample redundancy under tight memory constraints. While class-guided baselines like PBRS and CSTU regulate class occupancy, they do not prevent near-duplicate samples from consuming multiple slots. By enforcing a within-class diversity criterion, GOTTA enables the bounded memory to cover a wider range of predictive distributions. Although memory is not universally optimal SAR and EATA perform best without memory, while NOTE favors DAB-buffer, GOTTA's consistent gains across most objectives demonstrate that diversity-aware sample retention is an effective general strategy. Mean and standard deviation results across three random seeds are provided in Appendix A.5.

4.2 Generalization to ImageNet-C* and ITD

To evaluate the behavior of memory policies beyond the CIFAR-10-C setting, we use a fixed 10-class subset of ImageNet-C, denoted ImageNet-C*, under a memory budget of $M = 64$, with Dirichlet concentration settings $\gamma = 10^{-1}$ and $\gamma = 10^{-4}$; for the video-based ITD benchmark, we use only $\gamma = 10^{-1}$. We restrict ImageNet-C* to 10 classes – *goldfish, Robin, golden retriever, tabby cat, zebra, ant, dragonfly, candle, toilet tissue,* and *potpie* rather than the full 100, since with $M = 64$, 100 classes would yield under one sample per class on average, leaving no room to assess diversity; 10 classes guarantee over 6 slots per class, ensuring each can hold at least two instances, the minimum needed for a fair diversity comparison. The ITD benchmark contains 21 classes.

Table 2: Memory-equipped episodic test-time adaptation on ImageNet-C* (10 classes) and ITD. Results are reported with $M = 64$. ImageNet-C* is evaluated under two sampler settings, $\gamma = 10^{-1}$ and $\gamma = 10^{-4}$, whereas ITD is evaluated with $\gamma = 10^{-1}$. **Bold** denotes the best result in each column, while underlining denotes the second-best result.

		ImageNet-C*, $\gamma = 10^{-1}$			**ImageNet-C***, $\gamma = 10^{-4}$			**ITD**, $\gamma = 10^{-1}$		
Policy	Memory	RoTTA	NOTE	DAB	RoTTA	NOTE	DAB	RoTTA	NOTE	DAB
Uninformed	—	11.44	9.19	12.78	10.01	4.43	5.65	2.99	2.85	5.65
	Reservoir	12.12	22.87	**24.22**	11.64	**22.89**	**24.21**	3.56	4.40	11.96
Class-guided	PBRS	12.36	22.91	22.82	11.97	22.89	21.78	5.91	7.46	19.56
	CSTU	11.05	23.01	23.81	11.23	22.83	22.48	6.82	7.31	30.13
Diversity-aware	DAB-buffer	10.44	22.67	19.88	11.21	22.68	20.01	**34.44**	**25.38**	**43.91**
	GOTTA (OURS)	**13.82**	**23.09**	22.25	**14.25**	22.76	22.02	14.20	13.43	29.28

Table 2 summarizes these cross-benchmark evaluations. GOTTA improves performance in several challenging non-i.i.d. adaptation settings, although the magnitude of the benefit depends on the underlying adaptation method and dataset. On ImageNet-C*, GOTTA achieves the highest accuracy among the evaluated memory policies for RoTTA under both stream settings (13.82% at $\gamma = 10^{-1}$ and 14.25% at $\gamma = 10^{-4}$) and achieves the top score for NOTE at $\gamma = 10^{-1}$ (23.09%). On the temporally evolving ITD benchmark, incorporating GOTTA likewise delivers substantial gains, increasing RoTTA's performance from 2.99% to 14.20% and NOTE's from 2.85% to 13.43%. These results span two distinct non-i.i.d. regimes, Dirichlet-controlled class-imbalanced streams and temporally correlated video streams, highlighting that GOTTA's diversity-seeking mechanism remains effective across different stream structures and adaptation algorithms without requiring domain-specific retuning.

Task-specific designs such as DAB-buffer remain strong on ITD, showing the benefit of adapting memory selection to the data regime. At the same time, GOTTA provides a general-purpose alternative that remains effective across different datasets, stream structures, and adaptation methods. Overall, these results show that diversity-aware memory selection can improve test-time adaptation across diverse non-i.i.d. settings.

4.3 Ablation Studies

Effect of memory capacity. Table 3 studies how memory capacity affects the benefit of diversity-aware selection. GOTTA remains effective even under severe memory constraints: at the smallest budget of $M = 8$, it substantially improves over adaptation without memory, increasing accuracy from 24.74% to 56.58% for Norm and from 24.94% to 48.90% for DAB. As memory capacity increases, multiple samples can be retained per pseudo-class, giving the diversity mechanism more room to operate. This advantage becomes particularly evident at $M = 32$, where GOTTA achieves the highest accuracy across all three adaptation methods. At $M = 64$, it continues to perform best for RoTTA and DAB while remaining competitive for Norm. Overall, the capacity sweep shows that GOTTA is effective even with highly constrained memory, while its diversity-aware benefit becomes more pronounced as sufficient within-class capacity becomes available.

Effect of the diversity threshold. Our diversity criterion operates on softmax predictions $z \in \Delta^{C-1}$ sharing a pseudo-class. Since predictive distributions lie on the probability simplex and satisfy $0 \leq |z_i - z_j|_2 \leq \sqrt{2}$, we evaluate $\epsilon \in 0, 0.25, 0.5, 0.75, 1.0, 1.414$. Here, $\epsilon = 0$ imposes no diversity constraint, while larger values require greater predictive separation within the same pseudo-class.

Figure 2 reveals three distinct response patterns across CIFAR-10-C and ImageNet-C*:

Threshold-sensitive methods (Norm, RoTTA, DAB) perform best under permissive thresholds ($\epsilon \leq 0.25$) and degrade as the diversity constraint becomes stricter. On CIFAR-10-C, Norm and RoTTA drop from ∼74% to 50% at $\epsilon = 1.0$, while DAB drops from 76% to 47%. On ImageNet-C*, DAB declines from 22% to 16.5%, while RoTTA drops from 14% to 6.5%. These results indicate that aggressive diversity filtering can be detrimental to methods that benefit from retaining a broader set of observations.

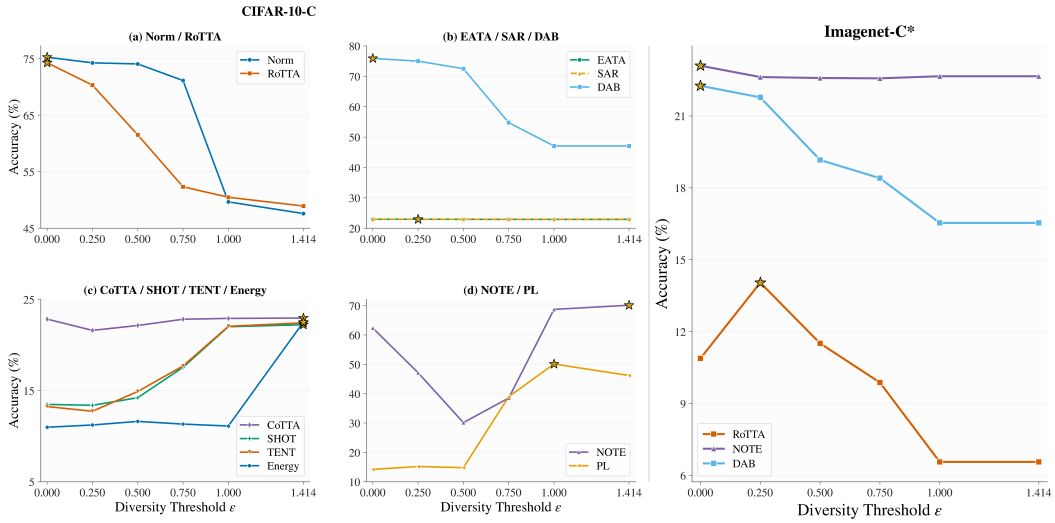

Figure 2: Effect of the diversity threshold ϵ on GOTTA. Accuracy is reported across TTA methods on **CIFAR-10-C** (left) and **ImageNet-C*** (right). We use a common set of candidate thresholds across methods to characterize their sensitivity to the strength of the diversity constraint. Results are reported using seed 42 and with $\gamma = 10^{-1}$.

Table 3: Memory-capacity scaling on **CIFAR-10-C** under temporal continual test-time adaptation. Rows report individual memory mechanisms grouped by memory-policy family, while columns report RoTTA, NORM, and DAB across memory budgets $M \in \{8, 16, 32, 64\}$.

Policy	Memory	RoTTA				NORM				DAB			
		$M=8$	$M=16$	$M=32$	$M=64$	$M=8$	$M=16$	$M=32$	$M=64$	$M=8$	$M=16$	$M=32$	$M=64$
Uninformed	—	50.14	50.14	50.14	50.14	24.74	24.74	24.74	24.74	24.95	24.94	24.94	24.94
	Reservoir	17.54	27.36	40.72	57.72	30.54	37.70	50.50	64.89	28.87	36.32	48.85	64.30
Class-guided	PBRS	**39.19**	**57.23**	69.10	75.67	**62.89**	67.80	72.50	**76.58**	**58.02**	65.69	71.96	74.54
	CSTU	33.03	55.33	69.21	75.23	51.75	62.61	71.49	75.45	48.18	61.60	71.36	75.47
Diversity-aware	DAB-buffer	31.70	49.65	57.92	52.04	57.15	65.42	59.91	52.61	51.09	62.08	59.02	52.39
	GOTTA (OURS)	34.47	55.35	**74.54**	**76.97**	56.58	**70.91**	**75.90**	75.34	48.90	**66.59**	**75.87**	**76.59**

Redundancy-sensitive methods instead benefit from stricter filtering. On CIFAR-10-C, Pseudo-Labeling (PL) improves from 14% to 50% at $\epsilon = 1.0$, while SHOT, TENT, Energy, and NOTE also show substantial gains. For these methods, removing near-duplicate predictions appears to reduce redundant signals in the retained memory and improve adaptation.

Relatively threshold-invariant methods show substantially smaller changes across ϵ. EATA, SAR, and CoTTA on CIFAR-10-C, as well as NOTE on ImageNet-C*, remain comparatively stable across the evaluated thresholds. This suggests that their internal sample selection or stabilization mechanisms may already mitigate some of the redundancy targeted by external diversity filtering.

5 Discussion and Conclusion

We revisited the role of memory in test-time adaptation (TTA), showing that memory should be treated as an actively curated coreset rather than a passive buffer. Recency-based and class-guided strategies can retain redundant samples under temporally correlated test streams, even when they control class representation. In contrast, GOTTA combines class coverage with within-class diversity through FPS, reducing redundancy among retained samples. Across our experiments, GOTTA remains effective across memory budgets and TTA methods, with its benefits becoming more apparent as memory capacity allows within-class diversity to influence selection. These results highlight memory selection, rather than memory size alone, as an important design dimension for memory-based TTA. While class-aware allocation depends on pseudo-label accuracy, our dynamic variant GOTTA-D provides a controlled analysis of periodically refreshing pseudo-label assignments.

AI DISCLOSURE

Required Disclosure. In this work, we **did not use** generative AI tools for any of the research tasks requiring disclosure, including generating synthetic datasets, developing theoretical or conceptual models, formulating mathematical claims or proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, interpreting results, or conducting data analysis.

Recommended Disclosure. Generative AI tools were used to edit and polish the writing of the manuscript, including improving readability, clarity, grammar, and phrasing. We also used a generative AI tool to create the teaser image presented in the paper, based on the concepts and text developed by the authors. Generative AI tools were not used to generate scientific results, design experiments, implement methods, analyze or interpret results, or develop the research methodology. All AI-assisted text and the generated image were reviewed by the authors, who take full responsibility for the final content of the manuscript and the presented figures.

REFERENCES

Shyma Alhuwaider, Motasem Alfarra, Juan C. Perez, Merey Ramazanova, and Bernard Ghanem. Advmem: Adversarial memory initialization for realistic test-time adaptation via tracklet-based benchmarking, 2025. URL https://arxiv.org/abs/2509.02182.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, pp. 11816–11825, 2019.

Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *CVPR*, pp. 8344–8353, 2022.

Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pp. 233–248, 2018.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in neural information processing systems*, 31, 2018.

Mario Döbler, Florian Marencke, Robert A. Marsden, and Bin Yang. Diversity-aware buffer for coping with temporally correlated data streams in online test-time adaptation. *arXiv preprint arXiv:2401.00989*, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *IEEE transactions on image processing*, 6(9):1305–1315, 1997.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016.

Silvio Giancola, Anthony Cioppa, Julia Georgieva, Johsan Billingham, Andreas Serner, Kerry Peek, Bernard Ghanem, and Marc Van Droogenbroeck. Towards active learning for action spotting in association football videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 5098–5108, June 2023.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Robust continual test-time adaptation: Instance-aware BN and prediction-balanced memory. In *NeurIPS*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pp. 448–456, 2015.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *NeurIPS*, pp. 2427–2440, 2021.

Minguk Jang and Sae-Young Chung. Test-time adaptation via self-training with nearest neighbor information. *CoRR*, abs/2207.10792, 2022.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. URL http://arxiv.org/abs/1612.00796.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, December 2012. ISSN 1935-8245. doi: 10.1561/2200000044. URL http://dx.doi.org/10.1561/2200000044.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3366–3385, 2022. doi: 10.1109/TPAMI.2021.3057446. URL https://doi.org/10.1109/TPAMI.2021.3057446.

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.

Shuang Li, Binhui Xie, Qiuxia Lin, Chi Harold Liu, Gao Huang, and Guoren Wang. Generalized domain conditioned adaptation network. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(8):4093–4109, 2022.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, pp. 6028–6039, 2020.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Transferable representation learning with deep adaptation networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(12):3071–3085, 2019.

Zachary Nado, Shreyas Padhy, D. Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *CoRR*, abs/2006.10963, 2020.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, volume 162, pp. 16888–16905, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world, 2023. URL https://arxiv.org/abs/2302.12400.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. 2008.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pp. 5533–5542, 2017.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 3723–3732, 2018.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS 2020*, 2020.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, pp. 9229–9248, 2020.

Jeffrey S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, March 1985. ISSN 0098-3500. doi: 10.1145/3147.3165. URL https://doi.org/10.1145/3147.3165.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153, 2018.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, pp. 7191–7201, 2022.

Binhui Xie, Shuang Li, Mingjia Li, Chi Harold Liu, Gao Huang, and Guoren Wang. Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–17, 2023.

Yongcan Yu, Lijun Sheng, Ran He, and Jian Liang. Stamp: Outlier-aware test-time adaptation with stable memory replay, 2024. URL https://arxiv.org/abs/2407.15773.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios, 2023. URL https://arxiv.org/abs/2303.13899.

Yige Yuan, Bingbing Xu, Liang Hou, Fei Sun, Huawei Shen, and Xueqi Cheng. Tea: Test-time energy adaptation, 2024. URL https://arxiv.org/abs/2311.14402.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pp. 289–305, 2018.

A Appendix

A.1 Memory Organization

The memory has a fixed global capacity M and is organized into pseudo-label partitions $\{\mathcal{M}^{(c)}\}_{c=1}^{C}$. Each incoming sample is assigned to the partition corresponding to its current pseudo-label $\hat{y}$. To encourage class coverage, each partition is assigned a target capacity

$$M_c = \left\lceil \frac{M}{C} \right\rceil,$$

where C is the number of classes. When $M < C$, the memory cannot allocate a slot to every class; in this case, the global capacity constraint limits the number of occupied class partitions to at most M at any time. The global memory constraint is enforced independently such that

$$\sum_{c=1}^{C} |\mathcal{M}^{(c)}| \leq M.$$

When a partition reaches its target capacity, the corresponding memory policy determines whether the incoming sample should be discarded or replace an existing entry.

A.2 ITD Setup

The ITD experiments use the following configuration:

- **Number of Classes:** 21 classes.
- **Image resolution:** 224×224 with 3 input channels.
- **Number of examples:** 10,000.
- **Corruption severity:** 5.
- **Corruption order:** order 0, following the corruption sequence defined in Appendix A.3.
- **Model:** ResNet-18.
- **Optimization batch size:** 64.
- **Stream length:** 32.

A.3 PTTA Stream Construction

For all experiments, to limit the computational cost, we use the same fixed corruption order across all adaptation methods and memory policies. Specifically, the corruptions are presented in the following order: Specifically, order 0 following (Yuan et al., 2023) corresponds to: Brightness→Pixelate→Gaussian Noise→Motion Blur→ Zoom Blur→Glass Blur→Impulse Noise→JPEG Compression→Defocus Blur→ Elastic Transform→Shot Noise→Frost→Snow→Fog→Contrast.

Within each corruption, we follow the PTTA protocol to construct temporally correlated non-i.i.d. streams, with class proportions controlled by a Dirichlet distribution. We use $\gamma = 10^{-1}$ and $\gamma = 10^{-4}$. The same stream construction is used across all compared memory policies to ensure that differences arise from the memory-selection mechanism rather than from different test-stream realizations.

A.4 Test-Time Adaptation settings

Following Yuan et al. (2023), we distinguish four TTA settings of increasing difficulty, all sharing the same protocol: a source-trained model f_{θ_s} is adapted online using only sequential unlabeled batches $\{\mathcal{X}_t\}_{t=0}^{T}$.

- **Fully Test-Time Adaptation (TTA)** (Wang et al., 2021). The target distribution $\mathcal{P}_t$ is stationary and batches are i.i.d. The goal is a one-shot adaptation to a fixed shift $\mathcal{P}_t \neq \mathcal{P}_s$.

Table 4: Memory-constrained comparison on **CIFAR-10-C** in continual temporal streams with $M = 32$ for all memory-based methods. Results are reported as mean $\pm$ standard deviation across three random seeds $\{42, 3407, 12345\}$. We mark — for no-memory baseline.

Policy	Memory	Source	Energy	SHOT	TENT	SAR	EATA	CoTTA	PL	NOTE	Norm	RoTTA	DAB
Uninformed	—	**56.49 ± 0.00**	12.29 ± 0.21	16.48 ± 0.44	15.55 ± 0.61	**24.93 ± 0.14**	**24.80 ± 0.12**	22.50 ± 0.21	13.78 ± 1.51	22.88 ± 0.30	24.80 ± 0.08	50.21 ± 0.53	25.01 ± 0.13
	FIFO	–	10.89 ± 0.64	12.44 ± 0.26	12.20 ± 0.53	23.15 ± 0.06	22.98 ± 0.06	20.87 ± 0.05	11.33 ± 0.52	43.17 ± 0.98	67.67 ± 0.21	64.16 ± 0.25	68.41 ± 0.27
	Reservoir	–	10.89 ± 0.25	13.10 ± 0.68	12.29 ± 0.40	23.19 ± 0.09	23.00 ± 0.03	21.84 ± 0.04	10.45 ± 0.30	52.30 ± 1.66	50.49 ± 0.45	40.89 ± 0.59	49.08 ± 0.52
Class-guided	PBRS	–	11.32 ± 0.14	13.76 ± 0.21	14.98 ± 0.65	22.86 ± 0.10	23.00 ± 0.03	21.58 ± 0.06	15.10 ± 0.87	**69.97 ± 0.68**	72.60 ± 0.10	68.89 ± 0.36	71.48 ± 0.43
	CSTU	–	10.96 ± 0.12	15.91 ± 0.30	15.11 ± 0.08	22.99 ± 0.03	22.99 ± 0.04	22.13 ± 0.07	13.99 ± 0.84	69.91 ± 0.52	71.41 ± 0.15	69.62 ± 0.40	71.24 ± 0.12
Diversity-aware	DAB-buffer	–	12.06 ± 0.26	15.03 ± 0.28	17.37 ± 0.81	22.94 ± 0.07	22.97 ± 0.04	22.35 ± 0.05	18.93 ± 5.51	70.95 ± 0.26	59.49 ± 0.37	57.99 ± 0.23	58.30 ± 0.66
	GOTTA (OURS)	–	**22.23 ± 0.27**	**22.25 ± 0.11**	**22.45 ± 0.15**	22.99 ± 0.03	22.99 ± 0.03	**22.99 ± 0.03**	**41.55 ± 5.71**	68.72 ± 1.53	**75.41 ± 0.13**	**75.66 ± 0.45**	**76.06 ± 0.18**

- **Continual Test-Time Adaptation (CoTTA)** (Wang et al., 2022). The target distribution drifts as a sequence $\{\mathcal{P}_t\}$, while batches remain i.i.d. within each $\mathcal{P}_t$. The challenge is tracking distributional drift without forgetting.
- **Non-i.i.d. Test-Time Adaptation** (Boudiaf et al., 2022; Gong et al., 2022). The target distribution is stationary, but class proportions within each batch are skewed by a Dirichlet distribution $\mathrm{Dir}(\gamma)$. A small concentration parameter γ produces highly imbalanced streams where certain classes dominate.
- **Practical Test-Time Adaptation (PTTA)** (Yuan et al., 2023). The most realistic setting: $\mathcal{P}_t$ evolves over time *and* batches are class-imbalanced via $\mathrm{Dir}(\gamma)$, jointly capturing temporal drift and label skew.

We follow PTTA setting in this work.

A.5 Results with Standard Deviations

To assess the robustness of the reported results, we additionally report the mean and standard deviation across three random seeds $42, 3407, 12345$ in Table 4. All memory-based methods use the same memory budget of $M = 32$ in continual temporal streams , enabling a direct comparison of both average performance and variability across runs. The results are consistent with the trends observed in the main experiments: the diversity-aware GOTTA memory generally achieves strong performance across adaptation methods, while the standard deviations are typically small, indicating stable behavior across random seeds. For completeness, we report the full per-method variability rather than only the mean values.

A.6 Robustness to Pseudo-Label Errors

Since GOTTA uses pseudo-labels to partition the memory, incorrect pseudo-labels may assign samples to incorrect class partitions and subsequently affect memory selection. To examine the effect of periodically revisiting these assignments, we introduce a dynamic variant, denoted as **GOTTA-D**. The results are reported in Table 5.

Table 5: Pseudo-label robustness comparison on **CIFAR-10-C** under PTTA. GOTTA uses the standard FPS memory policy, while GOTTA-D dynamically refreshes pseudo-label assignments and memory every three batches using FPSD. The best result for each adaptation method is highlighted in bold.

Memory Policy	NORM	RoTTA	DAB	NOTE	PL	TENT	SHOT	Energy	CoTTA	ETA	EATA	SAR
GOTTA-D	75.27	74.28	73.60	68.73	**50.11**	22.04	22.00	11.62	22.91	**22.96**	**22.96**	**22.97**
GOTTA	**76.21**	**75.42**	**75.87**	**70.48**	46.18	**22.43**	**22.32**	**22.50**	**22.95**	**22.96**	**22.96**	22.96

In the standard **GOTTA** setting, samples are assigned to memory partitions according to their current pseudo-labels and selected using FPS. In **GOTTA-D**, we periodically revisit the samples stored in memory. Specifically, after every three batches, we recompute the pseudo-labels of all retained samples using the current model predictions, reassign them to partitions according to their updated pseudo-labels, and reconstruct the memory using the FPS selection criterion. Thus, samples that were initially assigned to an incorrect partition can be reconsidered as the model predictions evolve. The comparison shows that the effect of this dynamic refresh depends on the adaptation method.

Table 6: Ablation of the CSTU-style eviction heuristic in GOTTA on **CIFAR-10-C**. Values report the mean accuracy (%) across the evaluated diversity thresholds $\epsilon \in \{0, 0.25, 0.5, 0.75, 1.0\}$. GOTTA (FPS only) uses the proposed FPS diversity criterion without the CSTU heuristic, while GOTTA (FPS + CSTU) uses the complete proposed configuration. CSTU is included as the baseline. Bold denotes the highest accuracy within each column.

Memory Policy	CoTTA	DAB	EATA	Energy	ETA	Norm	NOTE	PL	RoTTA	SAR	SHOT	TENT
CSTU	22.06	71.02	22.94	11.04	22.94	71.49	69.46	14.94	69.21	22.96	16.25	15.15
GOTTA (FPS only)	22.76	65.46	22.94	**16.15**	22.94	65.42	60.95	26.71	59.74	22.96	17.87	17.41
GOTTA (FPS + CSTU) (Ours)	**22.72**	**73.46**	**22.94**	15.99	**22.94**	**69.40**	**66.63**	**28.22**	**62.22**	**22.96**	**17.93**	**17.65**

Table 7: Memory-equipped episodic test-time adaptation on **ImageNet-C**. Experiments use a ResNet-50 backbone and report episodic temporal adaptation with $M = 64$ under two sampler settings, $\gamma = 10^{-1}$ and $\gamma = 10^{-4}$.

		ImageNet-C, $\gamma = 10^{-1}$			**ImageNet-C**, $\gamma = 10^{-4}$		
Policy	Memory	RoTTA	NOTE	DAB	RoTTA	NOTE	DAB
Uninformed	—	30.79	11.88	34.67	20.54	2.07	3.80
	Reservoir	31.66	10.12	41.03	27.54	10.87	32.78
Class-guided	PBRS	29.86	13.74	43.47	20.87	8.99	29.25
	CSTU	**38.59**	12.48	**45.85**	**32.85**	5.46	**34.97**
Diversity-aware	DAB-buffer	14.83	**16.98**	23.80	13.04	**13.70**	20.85
	GOTTA (OURS)	37.88	13.25	44.04	32.62	8.88	34.86

GOTTA-D substantially improves the performance of PL, increasing accuracy from 46.18% to 50.11%, while the differences for ETA, EATA, and SAR are negligible. For several other adaptation methods, including NORM, RoTTA, DAB, NOTE, TENT, SHOT, Energy, and CoTTA, GOTTA obtains higher accuracy than GOTTA-D. These results suggest that periodically refreshing pseudo-label assignments can alleviate the effect of stale or incorrect assignments in some adaptation settings, but the benefit is not uniform across methods.

Overall, this controlled experiment indicates that pseudo-label errors are not necessarily eliminated by dynamic refreshing, but that periodically revisiting the assignments provides a mechanism for correcting samples whose predicted class changes over time. Importantly, the refresh procedure uses only the model's current predictions and does not access ground-truth labels.

A.7 Ablation of the CSTU Eviction Heuristic

We further examine the role of the CSTU-style eviction heuristic in GOTTA. We compare GOTTA (FPS only), which applies the FPS diversity criterion without the CSTU heuristic, with GOTTA (FPS + CSTU), our complete configuration. The CSTU baseline is also included for reference. Table 6 reports the mean accuracy across the five evaluated diversity thresholds $\epsilon \in \{0, 0.25, 0.5, 0.75, 1.0\}$.

The results show that the FPS diversity criterion provides a strong contribution independently of the eviction heuristic, with GOTTA (FPS only) improving substantially over the CSTU baseline for several adaptation methods. Adding the CSTU heuristic further improves the average performance for several objectives, with particularly notable gains for DAB, NOTE, PL, and RoTTA. The effect is smaller for EATA, ETA, and SAR, while CoTTA and Energy show only minor changes.

Overall, the combined configuration improves the average accuracy across the evaluated adaptation methods compared with FPS alone. This suggests that the CSTU-style heuristic complements FPS by providing a criterion for selecting which stored sample to evict when the memory is full, while FPS controls redundancy among samples assigned to the same pseudo-class. We therefore use GOTTA (FPS + CSTU) as the complete configuration in the main experiments.

Table 8: Memory-equipped episodic test-time adaptation on **ImageNet-C** using a **Vision Transformer (ViT) backbone**. Results are reported with a memory budget of $M = 64$ under two sampler settings, $\gamma = 10^{-1}$ and $\gamma = 10^{-4}$.

		ImageNet-C, $\gamma = 10^{-1}$			**ImageNet-C**, $\gamma = 10^{-4}$		
Policy	Memory	RoTTA	NOTE	DAB	RoTTA	NOTE	DAB
Uninformed	—	54.35	53.84	54.61	**54.36**	**54.21**	53.24
	Reservoir	52.99	42.22	41.94	54.72	53.23	54.92
Class-guided	PBRS	54.47	**58.42**	**62.05**	52.03	53.72	54.57
	CSTU	55.07	57.86	61.23	51.76	50.68	55.24
Diversity-aware	DAB-buffer	54.71	54.01	53.81	53.15	53.04	52.47
	GOTTA (OURS)	**55.80**	56.73	59.41	52.75	51.59	**56.79**

A.8 Full ImageNet-C Evaluation

To further examine memory selection beyond the 10-class ImageNet-C subset used in the main experiments, we additionally evaluate on the full 1,000-class ImageNet-C benchmark. This setting raises an important fairness consideration for memory-based selection: under the fixed memory budgets used in our experiments, it is impossible to maintain comparable class coverage across 1,000 classes. For example, retaining only two samples per class would require a memory capacity of 2,000, whereas our largest memory contains substantially fewer samples. Thus, methods that explicitly promote class-wise coverage are inherently constrained by the available capacity, and the full 1,000-class setting should not be interpreted as a balanced comparison of per-class memory coverage. Instead, we use it as a stress test of memory selection when the number of competing classes greatly exceeds the available memory capacity. As shown in Table 7, GOTTA continues to improve over the uninformed no-memory setting across the evaluated adaptation methods and sampler settings, indicating that its benefits persist even under this highly constrained regime.

A.9 Generalization Across Architectures on ImageNet-C

We further investigate whether the behavior of different memory policies generalizes across model architectures in table 8. In addition to the ResNet-based experiments, we evaluate the memory mechanisms using a Vision Transformer (ViT) backbone on ImageNet-C. This experiment isolates the effect of the memory policy from the underlying architecture and examines whether the observed trends persist when moving from convolutional networks to transformer-based models.

Computational complexity of memory policies. We analyze the computational cost of the memory-selection policies considered in our experiments. Complexity is measured per incoming batch of size B, assuming that model predictions, pseudo-labels, uncertainty estimates, and predictive distributions have already been computed. Let M denote the memory capacity, C the number of classes, and $m = |\mathcal{M}(\hat{y}_t)|$ the number of retained samples in the pseudo-class associated with an incoming sample, where $m \leq M$.

The resulting asymptotic complexities are summarized in Table 9. Reservoir sampling requires constant-time updates, while PBRS maintains class-level occupancy information, giving $O(C)$ complexity. CSTU may examine the stored memory when selecting a sample for removal, resulting in $O(M)$ worst-case complexity. DAB compares incoming predictive distributions against a C-dimensional reference, giving $O(BC)$ per batch.

GOTTA performs nearest-neighbor comparisons in predictive-distribution space only within the relevant pseudo-class. For each incoming sample, it compares its C-dimensional predictive distribution with m retained samples, giving $O(mC)$ complexity. Thus, for a batch of B samples, the complexity is $O(BmC)$, with a worst-case bound of $O(BMC)$. Under approximately balanced pseudo-class occupancy, $m \approx M/C$, giving a typical comparison cost of approximately $O(BM)$. This additional cost reflects the explicit within-class diversity constraint introduced by GOTTA.

Table 9: Asymptotic computational complexity of the memory-selection mechanisms. Complexity is measured per incoming batch of size B, assuming that model predictions, pseudo-labels, uncertainties, and predictive distributions have already been computed. M denotes the total memory capacity, C the number of classes, and $m \leq M$ the number of retained samples in the relevant pseudo-class.

Memory Policy	**Complexity**
None	$O(1)$
Reservoir	$O(1)$
PBRS	$O(C)$
CSTU	$O(M)$
DAB	$O(BC)$
GOTTA (Ours)	$O(BmC)$