GOTTA BE DIVERSE

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-Time Adaptation (TTA) enables models to adjust to distribution shifts using only the incoming test stream. While existing methods perform well under covariate shifts, their performance drops when label distributions also change, a common scenario in real-world streams. Some approaches attempt to mitigate this by introducing memory modules into their methods, typically to enforce class balance. However, because these memories are evaluated only in conjunction with specific algorithms, their independent role and effectiveness remain unclear. In this work, we systematically study memory in TTA by decoupling it from the adaptation algorithm. Through a unified evaluation, we identify the design choices that make memory effective under different stream settings. Building on these insights, we propose Guided Observational Test-Time Adaptation (GOTTA), a category of diversity-aware memories that combine class balance with intra-class diversity. Our results show that such memories provide reliable, compact, and efficient support for adaptation in dynamic test streams, highlighting diversity-aware memory as an important principle for robust TTA.

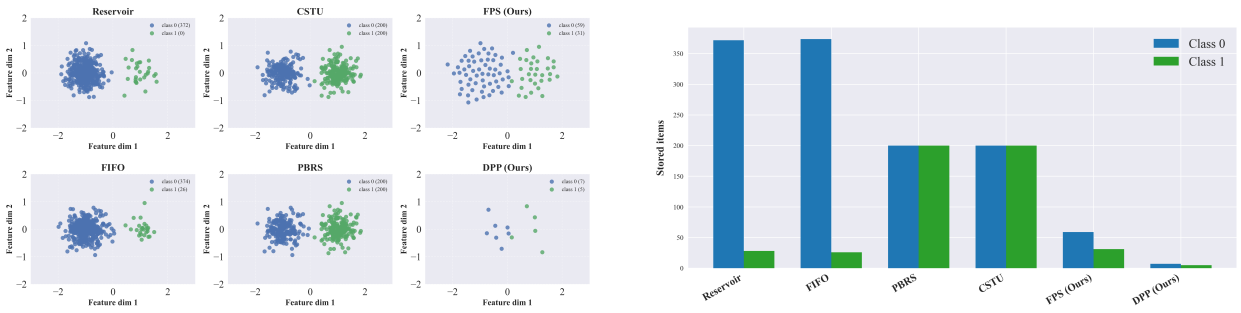
1 INTRODUCTION

Deep neural networks (DNNs) have achieved state-of-the-art performance on a wide array of tasks, fueled by large-scale datasets and powerful architectures (He et al., 2016; LeCun et al., 2015). A fundamental assumption underpinning this success is that training and testing data are independently and identically distributed (i.i.d.). However, in real-world deployments, this assumption is frequently violated. Models encounter streams of data whose distributions shift over time due to factors such as changing weather conditions, varying lighting, or deployment in new environments, a phenomenon known as dataset shift (Quinonero-Candela et al., 2008). Such shifts can cause severe drops in performance, limiting the reliability of AI systems. To ensure robustness, models must be able to adapt continuously after deployment. Among various paradigms such as Unsupervised Domain Adaptation (UDA) (Wang & Deng, 2018) and Domain Generalization (DG) (Zhou et al., 2022), Test-Time Adaptation (TTA) has emerged as a particularly practical approach, as it adapts online without requiring access to the source training data.

Test-Time Adaptation aims to adapt a pre-trained, source-free model using only an incoming stream of unlabeled test data. Early work addressed static shifts by updating batch normalization statistics (Schneider et al., 2020; Nado et al., 2020), minimizing prediction entropy (Wang et al., 2021), or leveraging pseudo-labeling (Lee et al., 2013) and self-supervision tasks such as rotation prediction (Sun et al., 2020) to cope with such shifts. While effective in simplified settings, these methods implicitly assume that test data is drawn from a single, stationary target domain. This assumption does not hold in practice, where data streams evolve continually and exhibit temporal redundancy.

The central challenge for TTA arises in **dynamic environments**, where test streams are non-stationary and non-i.i.d. Real-world examples, such as onboard vehicle cameras, combine gradual appearance shifts (e.g., sunny to rainy conditions) with shifts in class prevalence (e.g., long stretches of cars followed by pedestrians). In these settings, existing TTA methods struggle because they lack mechanisms to manage continual dynamic environment, which leads to error accumulation and forgetting. Recent studies such as RoTTA (Yuan et al., 2023) and NOTE (Gong et al., 2022) highlighted this issue and introduced memory buffers to stabilize adaptation. Their memory mechanisms primarily focus on managing capacity and enforcing class balance through heuristic scores.

In this work, we argue that **memory should be treated as a first-class, modular component of TTA**. Rather than a passive buffer, memory defines how well adaptation balances stability and plasticity. We therefore disentangle memory from the rest of the adaptation process and propose **diversity-aware memory modules** as a drop-in replacement for existing heuristics. Our key insight is that preserving diversity in memory is critical for approximating the evolving test distribution and preventing feature collapse. To this end, we instantiate GOTTA (*Guided Observational Test-Time Adaptation*) with two principled sampling strategies: *Farthest Point Sampling (FPS)*, which ensures broad feature-space coverage, and *Determinantal Point Processes (DPP)*, which probabilistically favor diverse and representative



(a) Feature space visualization.

(b) Final per-class memory occupancy.

Figure 1: **Comparison of memory mechanisms.** (a) Visualization of stored samples after many batches for two dummy classes (blue vs. green). *Uninformed memories* (FIFO, Reservoir (Vitter, 1985)) reflect the imbalanced data stream and suffer feature collapse, with minority classes poorly represented. *Class-guided memories* (CSTU(Yuan et al., 2023), PBRS (Gong et al., 2022)) enforce class balance but retain redundant samples. Our *GOTTA memories* (FPS, DPP) maintain both class balance and intra-class diversity by pruning redundant samples and preserving only informative ones. (b) Class occupancy statistics, showing imbalance for uninformed memories, balance but redundancy for class-guided memories, and balanced yet compact storage for our GOTTA memories.

subsets. This modular design allows GOTTA to be seamlessly integrated into virtually any Test-Time Adaptation method.

Contributions.

- We disentangle memory from adaptation and conduct a systematic benchmark of memory mechanisms across a wide range of TTA methods and stream settings, covering both i.i.d. and non-i.i.d. scenarios.
- Our unified evaluation reveals the design choices that most strongly affect memory effectiveness, including class balance, uncertainty, and intra-class diversity.
- Guided by these insights, we propose Guided Observational Test-Time Adaptation (GOTTA), a new category of diversity-aware memories. These mechanisms combine class balance with intra-class diversity, yielding more compact, reliable, and efficient buffers that support robust adaptation under challenging distribution shifts.

2 Related Work

2.1 Domain Adaptation

Our work is related to the broad field of Unsupervised Domain Adaptation (UDA), which addresses the challenge of domain shift by adapting a model from a labeled source domain to an unlabeled target domain (Mansour et al., 2009). Foundational UDA methods focused on aligning the feature distributions between domains to create a shared, domain-agnostic representation space. This is often achieved by minimizing statistical metrics like Maximum Mean Discrepancy (MMD) (Long et al., 2019), which compares the moments of the two distributions. A more prominent approach involves adversarial training, where a domain discriminator network is trained to distinguish between source and target features, while the main feature extractor is simultaneously trained to produce features that can fool this discriminator, thereby enforcing domain invariance (Ganin et al., 2016; Saito et al., 2018).

Another significant line of work involves self-training, where the model generates pseudo-labels on the target domain to retrain itself in a supervised fashion. However, this paradigm is vulnerable to the risk of error accumulation, where the model can become overconfident in its own incorrect predictions. To mitigate this, techniques like class-balancing, curriculum learning, and confidence thresholding are often incorporated to improve the quality and reliability of the generated pseudo-labels (Zou et al., 2018; 2019). While these UDA methods are powerful, they are fundamentally offline, requiring concurrent access to large batches of both source and target data during the adaptation phase. Our work, in contrast, operates in a more constrained and practical online, test-time setting where source data is unavailable.

2.2 Continual Learning

Continual Learning (CL) addresses the challenge of learning from a sequence of tasks or data streams while mitigating catastrophic forgetting—the tendency for a model to abruptly lose knowledge of previously learned tasks upon learning new ones (Kirkpatrick et al., 2016; Lange et al., 2022). The goal of CL in preventing forgetting aligns closely with the objectives of robust adaptation. Major CL strategies include regularization-based approaches, which penalize changes to weights deemed critical for past tasks (Kirkpatrick et al., 2016; Aljundi et al., 2019); parameter isolation methods, which assign distinct model parameters to different tasks; and rehearsal-based methods, which store a small subset of past data in an explicit memory buffer to be replayed during subsequent training (Castro et al., 2018; Rebuffi et al., 2017). This final strategy is most analogous to our work. However, CL typically operates on a sequence of distinct, labeled tasks. Our setting, Continual Test-Time Adaptation, presents a distinct challenge: the model must adapt to a single, continuous, and unlabeled stream of data with shifting distributions. Therefore, while we draw inspiration from the CL goal of preventing forgetting, our approach is tailored to the unique constraints of unsupervised, online adaptation where task boundaries are not explicitly defined.

2.3 Test-Time Adaptation: From Static Shifts to Continual Dynamics

Test-Time Adaptation (TTA) is constrained to a practical yet challenging online, source-free setting. Foundational methods were developed to handle adaptation to a single, **static distribution shift**. These approaches rely on varied adaptation signals, such as minimizing prediction entropy (Wang et al., 2021), leveraging self-supervised tasks (Sun et al., 2020), or aligning feature clusters with source statistics (Su et al., 2022). While effective for their intended purpose, these memory-less strategies are inherently ill-equipped for dynamic data streams, where they are prone to error accumulation and catastrophic forgetting.

This limitation revealed the need for Continual TTA (Wang et al., 2022), a paradigm designed for constantly changing, **dynamic environments**. To provide stability and historical context, the primary strategy to emerge has been the integration of a memory buffer. Several pioneering works have demonstrated that storing past test samples can anchor the adaptation process and mitigate forgetting (Gong et al., 2022; Chen et al., 2022; Yuan et al., 2024). However, the critical issue lies not in the use of memory itself, but in how that memory is managed. Current methods treat memory as a *passive reservoir* rather than an *active coreset*. Their management strategies remain heuristic, typically relying on simple recency (e.g., FIFO queues) or random chance (e.g., reservoir sampling) (Gong et al., 2022), or heuristic rule-based with class balance enforcement (Gong et al., 2022; Yuan et al., 2023). They lack a principled mechanism to assess the informational value of incoming data, thus may accumulate redundant samples while discarding unique ones. This results in a memory that fails to maintain a truly representative and diverse snapshot of the distribution. In our work, different from the previous literature, we introduce more efficient memory management, that increase diversity and reduce redundancy in the samples.

3 Methodology

3.1 Problem Definition

Let $f_{\theta_s} : \mathcal{X} \to \mathcal{Y}$ be a deep neural network pre-trained on a source dataset $\mathcal{D}_s = \{(x^s, y^s)\}$, where samples are drawn from a source distribution $\mathcal{P}_s$. In the Test-Time Adaptation (TTA) setting, we assume that at deployment time, we have access **only to the pre-trained model parameters** θ_s, with no access to the source data $\mathcal{D}_s$.

The model is then deployed into a real-world environment where it receives a continuous, unlabeled stream of test data $\mathcal{X}_0, \mathcal{X}_1, ..., \mathcal{X}_T$. This stream is drawn from a test distribution $\mathcal{P}_{test}$ that differs from the source distribution $\mathcal{P}_s$. This mismatch, known as **distribution shift**, causes a significant degradation in the performance of the static model f_{θ_s}.

The core challenge, which we address, arises in **dynamic, non-stationary environments**. In this practical scenario, the target distribution is not static but evolves over time. The data stream is thus drawn from a sequence of changing distributions $\{\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_t, \ldots\}$, where the distribution at time t, $\mathcal{P}_t$, may differ from the distribution at time $t-1$.

3.2 Test-Time Adaptation Settings

Following the categorization in (Yuan et al., 2023), we distinguish between several test-time adaptation (TTA) settings. In all cases, a source-trained model f_{θ_s} is adapted using only sequential batches $\mathcal{X}_0$, $\mathcal{X}_1$, $\mathcal{X}_2$, ..., $\mathcal{X}_T$ drawn from an unlabeled test stream distribution $\mathcal{P}_t$.

1. **Fully Test-Time Adaptation (TTA).** (Wang et al., 2021) The target distribution $\mathcal{P}_t$ is stationary, and batches $\mathcal{X}_t$ are sampled independently. The goal is to adapt once to this fixed shift $\mathcal{P}_t \neq \mathcal{P}_s$.
2. **Continual Test-Time Adaptation (CoTTA).** (Wang et al., 2022) The target distribution changes as a sequence $\{\mathcal{P}_t\}_{t=0}^{\infty}$, while batches remain i.i.d. within each $\mathcal{P}_t$. The challenge is to track distributional drift over time.
3. **Non-i.i.d. Test-Time Adaptation.** (Boudiaf et al., 2022; Gong et al., 2022) In this setting the target distribution $\mathcal{P}_t$ is stationary, but incorporates label imbalance induced by a Dirichlet distribution $\mathrm{Dir}(\gamma)$. A small concentration parameter γ produces highly skewed class proportions, creating streams where certain classes dominate despite $\mathcal{P}_t$ being fixed.
4. **Practical Test-Time Adaptation (PTTA).** (Yuan et al., 2023) PTTA generalizes this to non-stationary streams. Here $\mathcal{P}_t$ evolves over time while simultaneously encoding label imbalance through $\mathrm{Dir}(\gamma)$. The result is a test stream that is both dynamically shifting and imbalanced, closely mirroring real world deployment conditions.

Mathematical Objective of Adaptation. Across all stream settings, test-time adaptation can be formulated as an online optimization problem. At each time step t, the model receives a batch of unlabeled samples $\mathcal{X}_t$ and updates its parameters from θ_{t-1} to θ_t by minimizing an unsupervised loss $\mathcal{L}_{\text{unsup}}$, which acts as a surrogate for the supervised task loss:

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

Once adapted, the new weights θ_t is used to run inference on the next batch $\mathcal{X}_{t+1}$.

3.3 Test-Time Adaptation with Memory

To handle the challenges of non-i.i.d. data streams, such as catastrophic forgetting and sensitivity to temporally correlated data, the standard Test-Time Adaptation framework is often augmented with an external memory buffer, denoted as $\mathcal{M}$. The fundamental purpose of this memory is to maintain a rolling window of representative test samples, providing a more stable and approximately i.i.d. dataset to guide the adaptation process.

The Memory Module. The memory $\mathcal{M}$ is a fixed-capacity container of size $\mathcal{N}$. At time step t, it holds up to $\mathcal{N}$ entries from the test stream, $\mathcal{M}_t = \{m_1, m_2, \ldots, m_K\}$ with $K \leq \mathcal{N}$. Formally, each entry is $m_i = (x_i, \hat{y}_i, u_i, p_\theta(x_i), a_i)$, where x_i is the input, $\hat{y}_i$ its pseudo-label, u_i its entropy, $p_\theta(x_i)$ the probability embedding, and a_i the age. This design allows the memory to incorporate both semantic information (pseudo-labels and probability embeddings) and statistical cues (entropy and age), moving beyond simple sample storage.

Memory-Augmented Adaptation. At each time step t, when a new batch $\mathcal{X}_t$ arrives, adaptation proceeds in two stages:

1. **Memory Update:** The buffer is updated via a policy Π, i.e., $\mathcal{M}_t \leftarrow \Pi(\mathcal{M}_{t-1}, \mathcal{X}_t)$, where Π specifies how new samples are inserted and old ones evicted (e.g., FIFO, CSTU, FPS).
2. **Parameter Update:** Model parameters are adapted using the samples from the memory $\mathcal{M}_t$, $\theta_t \leftarrow \text{Adapt}(\theta_{t-1}, \mathcal{M}_t; \mathcal{L}_{\text{adapt}})$, where Adapt denotes a generic TTA objective (e.g., entropy minimization).

Memory-augmented TTA decouples parameter updates from noisy or correlated batches, leading to a more stable and robust adaptation trajectory compared to non-memory TTA.

Taxonomy of Memory Mechanisms. We categorize existing memory mechanisms for test-time adaptation into three main families, reflecting an increasing use of model-derived information: from ignoring model outputs, to utilizing pseudo-labels for class balance, to leveraging probability embeddings for intra-class diversity.

- **Uninformed Memories:** These mechanisms do not leverage pseudo-labels, features, or uncertainty. The most basic is *FIFO*, which simply discards the oldest sample, effectively reducing memory to a sliding window. *Reservoir Sampling* instead maintains a uniform random subset of all observed samples.
- **Class-Guided Memories:** These explicitly enforce class balance by partitioning memory across pseudo-labels. For example, *CSTU* applies a timeliness–uncertainty heuristic to evict items, while *PBRS* (Per-class Balanced Reservoir Sampling) extends reservoir sampling into a per-class formulation, ensuring minority classes are retained.

- **Guided Observational Memories (ours):** We introduce the GOTTA category , the first class of memories to explicitly combine *class balance* with *intra-class diversity*. Unlike prior categories, GOTTA mechanisms not only allocate memory fairly across classes but also ensure diverse coverage within each class in the probability embedding space. For instance, *FPS* enforces diversity via pairwise distance checks, while *DPP pruning* removes redundant items based on determinantal point process criteria.

This taxonomy highlights the progression from memories that ignore model outputs, to those guided by pseudo-labels for class balance, and finally to our proposed GOTTA memories that enforce both balance across classes and diversity within classes through probability embeddings.

Figure 1 compares memory families in a controlled two-class stream. We simulate a stream of $4{,}000$ samples from two classes with $p = (0.95, 0.05)$, a memory capacity of 400, and low-variance Gaussian features ($\sigma^2 = 0.1$). This stress-test amplifies imbalance and redundancy, exposing collapse in uninformed memories and contrasting behaviors across policies. Panel (a) illustrates stored samples: *uninformed memories* (FIFO, Reservoir) simply track the data stream, reproducing its class imbalance and collapsing minority-class features. *Class-guided memories* (CSTU, PBRS) achieve strict class balance, but retain highly redundant samples within each class. In contrast, *GOTTA memories* (FPS, DPP) selectively admit only informative samples, reducing redundancy and mitigating imbalance through intra-class diversity. The resulting buffers are compact and representative rather than saturated with near-duplicates. Panel (b) quantifies these trends: uninformed memories remain imbalanced, class-guided memories are balanced but redundant, and GOTTA memories strike a middle ground that preserves both balance and diversity.

3.4 GOTTA: Guided Observational Memories

We move beyond the limitations of heuristic-based memory management by introducing **GOTTA**, a category of *Guided Observational Memories* built on the central hypothesis that a memory buffer preserving both **class balance** and **intra-class diversity** provides a more robust and unbiased foundation for adaptation. Unlike *Class-Aware Memories* such as CSTU (Category-balanced sampling with timeliness and uncertainty) (Yuan et al., 2023), which passively accumulates recent or high-confidence samples, GOTTA's memory module, $\mathcal{M}$, functions as an actively curated coreset of the observed data stream.

Farthest Point Sampling (FPS). Our Π_{FPS} policy enforces both class balance and intra-class diversity. Each new item $m_t = (x_t, \hat{y}_t, u_t, p_\theta(x_t), a_t)$ is admitted through the following procedure:

1. **Capacity Handling:** We first check whether the class partition $\mathcal{M}^{(\hat{y}_t)}$ has free slots. If it is not full, the item can be considered for insertion. If the class is at capacity, we identify the nearest stored item m_r in the embedding space and compare m_t against m_r using the heuristic score

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

   where a_i is the item's age and u_i its entropy. The item with the lower score is retained, preferring younger and more confident samples, similar to previous work (Yuan et al., 2023).

2. **Diversity Check:** If the capacity step allows m_t to be inserted, we verify that it adds sufficient diversity within its class. Specifically, we compute the Euclidean distance between $p_\theta(x_t)$ and all stored embeddings in $\mathcal{M}^{(\hat{y}_t)}$; if the minimum distance exceeds a fixed threshold $\epsilon = 10^{-6}$, the item is added, otherwise it is discarded as redundant. For simplicity, we use this same threshold for all experiments without tuning.

In practice, we adopt a greedy variant of Farthest Point Sampling that avoids repeatedly computing global pairwise distances. This more efficient procedure preserves the same core behavior of FPS while greatly reducing computational overhead. For consistency, we will continue to refer to this method as **FPS** throughout the paper.

In this way, FPS maintains class balance while actively discouraging redundancy in the probability embedding space. Unlike FIFO or reservoir sampling, which admit items indiscriminately, FPS curates the buffer to prioritize younger, more confident, and diverse samples. Compared to class-aware memories such as CSTU and PBRS, which enforce balance but admit many near-duplicates, FPS explicitly filters for intra-class diversity, leading to a more compact and representative memory.

FPS with Dynamic Probability Embedding Updates (FPSD). While FPS relies on probability embeddings computed at the time of admission, these embeddings become stale as the model adapts and its parameters θ evolve. To

address this drift, FPSD periodically refreshes stored embeddings. Specifically, after every τ adaptation steps, we forward all items $x_i \in \mathcal{M}$ through the current model and update their embeddings $p_\theta(x_i)$ and entropies u_i. This ensures that diversity checks and heuristic comparisons are always evaluated in the most up-to-date feature space. The update does not alter the memory selection policy itself, but keeps the buffer aligned with the evolving model.

Determinantal Point Process Pruning (DPP). Our Π_{DPP} policy extends CSTU-style class balancing by incorporating diversity through determinantal point processes (DPPs). Each new item $m_t = (x_t, \hat{y}_t, u_t, p_\theta(x_t), a_t)$ is considered for insertion into its class partition $\mathcal{M}^{(\hat{y}_t)}$ as follows:

1. **Capacity Handling:** If the class partition has not reached its allocated capacity, m_t is added directly. If it is full, we construct a candidate set $\mathcal{M}^{(\hat{y}_t)} \cup \{m_t\}$ and proceed to pruning.
2. **Approximate Diversity Check:** For efficiency, we randomly subsample from the class partition and compare the normalized embeddings using cosine similarity. If the new item is too similar to existing ones (maximum similarity above $1 - \epsilon$, with $\epsilon = 0.1$), it is discarded as redundant. Otherwise, it is admitted to the candidate set.
3. **DPP-Based Pruning:** On the candidate set, we form a kernel

$$L_{ij} = \exp\left(-\frac{\|p_\theta(x_i) - p_\theta(x_j)\|_2^2}{\sigma^2}\right).$$

The log-determinant $\log\det(L)$ quantifies the joint diversity of the subset. We remove the single item whose exclusion maximizes $\log\det(L)$; if this is the new item m_t, it is rejected, otherwise it replaces the pruned element in memory.

In summary, DPP pruning goes further by casting diversity as a probabilistic selection problem. Unlike FIFO or Reservoir, which ignore labels and embeddings, and unlike CSTU or PBRS, which enforce balance but allow redundancy, DPP favors subsets with high log-determinant volume, ensuring diverse coverage of each class. This probabilistic formulation, combined with efficient subsampling, yields a compact yet informative memory.

In this way, GOTTA mechanisms balance class occupancy while promoting intra-class diversity, yielding more compact and representative memories. Further implementation details are provided in Appendix 6.2.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate our methods on CIFAR-10-C and CIFAR-100-C (Hendrycks & Dietterich, 2019). Each dataset consists of the clean test set corrupted by 15 types of common image corruptions, each applied at five levels of severity. For continual evaluation, we follow a fixed corruption order: [*brightness*, *pixelate*, *gaussian noise*, *motion blur*, *zoom blur*, *glass blur*, *impulse noise*, *jpeg compression*, *defocus blur*, *elastic transform*, *shot noise*, *frost*, *snow*, *fog*, *contrast*]. We evaluate all methods under two stream conditions: (i) an i.i.d. stream where samples are uniformly drawn, and (ii) a non-i.i.d. stream generated using the Practical TTA (PTTA) protocol (Gong et al., 2022; Yuan et al., 2023), which induces temporal correlations via Dirichlet sampling. Unless otherwise specified, we set the concentration parameter $\gamma = 10^{-1}$ for CIFAR-10C and $\gamma = 10^{-4}$ for CIFAR-100C.

Models. We adopt standard backbones of comparable capacity: WildResNet-28 (Zagoruyko & Komodakis, 2016) for CIFAR-10-C and ResNeXt-29 (Xie et al., 2017) for CIFAR-100-C.

Metrics. Our primary evaluation metric is top-1 classification accuracy averaged across the entire test stream, providing a comprehensive measure of overall performance.

TTA Baselines. We compare our GOTTA framework against a comprehensive suite of foundational and state-of-the-art Test-Time Adaptation methods. Our evaluation includes the non-adapting source model, classic approaches such as **TENT** (Wang et al., 2021), **SHOT** (Liang et al., 2020), and **PL** (Lee et al., 2013), and recent methods designed for continual and non-i.i.d. streams like **CoTTA** (Wang et al., 2022), **NOTE** (Gong et al., 2022), and **RoTTA** (Yuan et al., 2023). To ensure a thorough comparison, we also include also **EATA** (Niu et al., 2022), **ETA** (Niu et al., 2022), **SAR** (Niu et al., 2023), **Energy** (Yuan et al., 2024). All baseline adaptation methods, we adopt the hyperparameter settings reported in their respective original publications to ensure a fair and robust comparison.

Memory Modules. We evaluate our proposed memory curation policies against several foundational and state-of-the-art memory-augmented adaptation strategies. To ensure a fair comparison focused on the curation policy itself,

Table 1: Impact of memory curation policies on TTA methods (mean classification accuracy) across batch sizes.

Memory	Rotta					Norm					Note				
	8	12	16	32	64	8	12	16	32	64	8	12	16	32	64
NONE	—— 49.47 ——					—— 24.72 ——					—— 22.84 ——				
FIFO	63.48	53.15	63.81	64.09	55.72	52.15	63.55	52.78	54.41	64.78	71.02	67.08	69.90	63.61	58.48
RESERVOIR	60.30	55.36	60.48	63.57	60.69	56.62	58.56	57.90	59.72	63.36	68.03	66.85	69.33	69.13	68.50
CSTU	73.13	68.67	74.03	74.52	74.74	67.66	72.58	70.34	73.06	75.32	**74.03**	**73.68**	**73.92**	73.13	72.22
PBRS	**73.78**	69.54	75.28	75.34	75.60	69.22	73.61	72.18	73.59	**76.28**	72.13	72.94	72.59	72.01	70.12
DPP	73.13	68.95	**75.38**	**76.53**	73.60	67.76	**74.79**	71.06	73.48	75.36	74.02	73.20	73.68	69.42	64.05
FPS	73.15	**72.87**	75.02	75.36	**76.55**	68.12	74.48	71.58	**76.41**	76.09	74.01	73.37	73.87	**73.20**	**72.68**
FPSD	73.60	70.74	74.57	75.46	74.29	**70.31**	73.12	**72.72**	73.94	76.02	73.27	71.63	73.28	70.85	64.70

all memory-based methods use the same underlying adaptation objective of entropy minimization on the buffered samples.

- **FIFO Memory:** A foundational baseline where the memory buffer is a simple First-In-First-Out queue. When the memory is full, the oldest sample is evicted to make space for a new one. This policy exclusively prioritizes recency.
- **Reservoir Sampling Memory (Vitter, 1985):** This baseline uses standard Reservoir Sampling to maintain a uniform random sample of the entire stream observed so far. When a new sample arrives at step t ($t > K$), it is added to the memory with probability K/t, replacing a randomly chosen existing sample.
- **PBRS Memory (Gong et al., 2022):** The Prediction-Balanced Reservoir Sampling policy from the NOTE framework. It uses the model's pseudo-labels to achieve a class-balanced memory. It prioritizes replacing samples from majority classes and uses class-conditional reservoir sampling logic for non-minority classes.
- **CSTU Memory (Yuan et al., 2023):** The memory curation policy from the RoTTA framework. This is a heuristic-based policy that scores samples based on their age (timeliness) and prediction entropy (uncertainty), while also enforcing category-balance. Samples with higher heuristic scores (older, more uncertain) are preferentially evicted.
- **GOTTA (Ours):** Our proposed framework, representing principled diversity-aware curation policies.
  - **FPS**: Uses our Farthest Point Sampling policy.
  - **FPSD:** Uses our FPS with Dynamic Feature Updates policy.
  - **DPP**: Uses our Determinantal Point Process policy.

Finally, all adaptation methods are compared against the **Source model**, which is the pre-trained network evaluated directly on the test stream with no adaptation.

4.2 Main Results

The results presented in Table 1 demonstrate that the memory curation strategy is a critical determinant of performance in continual Test-Time Adaptation. Our experiments show that by replacing existing memory heuristics with our diversity-aware **GOTTA** policies (**DPP**, **FPS**, and **FPSD**), we consistently and substantially elevate the performance of various state-of-the-art TTA methods.

Elevating State-of-the-Art TTA Performance As shown in Table 1, our proposed policies achieve superior mean classification accuracy across all tested TTA methods. When augmenting the RoTTA, our **FPS** policy achieves a top accuracy of **76.55%**, a significant improvement over the 74.74% accuracy from RoTTA's native CSTU policy. This performance gain is consistent across different methods. For instance, when integrated with NOTE, our **FPS** policy (72.68%) again surpasses the method-specific PBRS policy (70.12%), highlighting the generalizability and effectiveness of our diversity-aware curation approach. Similarly, for the Norm method, our **FPSD** policy reaches **73.94%** accuracy, outperforming the best baseline policy, PBRS (73.59%).

Data Efficiency and Robustness to Memory Capacity A crucial advantage of the **GOTTA** is its ability to achieve state-of-the-art performance with far greater data efficiency. While baseline policies like CSTU and PBRS are highly

Table 2: Comparative analysis of memory curation policies under severe dataset shift. The table reports mean classification accuracy (%) on **CIFAR-10** → **CIFAR-10C**, averaged over 15 common corruptions at the highest severity level (5). All evaluations are conducted in a **continual** correlatively sampled stream with $\gamma = 10^{-1}$.

Policy	**Memory** \ **Method**	Source	CoTTA	EATA	Energy	ETA	Norm	NOTE	PL	RoTTA	SAR	SHOT	TENT
Uninformed	NONE	56.49	22.46	24.71	12.08	24.71	24.72	22.84	13.94	49.47	24.82	16.50	15.32
	FIFO	-	23.86	24.71	12.03	24.71	64.78	58.48	14.02	55.72	24.82	16.82	15.19
	Reservoir	-	24.51	24.71	11.94	24.71	63.36	68.50	**22.72**	60.69	24.74	**22.10**	**21.87**
Class-Guided	PBRS	-	23.92	24.71	12.44	24.71	**76.28**	70.12	20.01	75.60	**24.91**	16.18	18.02
	CSTU	-	24.16	24.69	11.88	24.69	75.32	72.22	21.95	74.74	24.72	16.06	18.12
Guided Observational	DPP (Ours)	-	24.16	**24.72**	**14.00**	**24.72**	75.36	64.05	12.53	73.60	24.72	16.33	19.96
	FPS (Ours)	-	**24.60**	24.71	12.36	24.71	76.09	**72.68**	14.69	**76.55**	24.72	14.72	16.48
	FPSD (Ours)	-	23.47	24.71	12.26	76.02	24.71	64.70	16.10	74.29	24.72	13.93	16.54

dependent on large memory buffers (peaking at 64 samples) to achieve their best results, our policies demonstrate remarkable robustness even at minimal capacities.

This efficiency is most striking when applying our policies to the RoTTA method. Our **DPP** policy achieves **76.53%** accuracy using a memory buffer of only 32 samples. This result not only surpasses the peak performance of the native CSTU policy (74.74% at 64 samples) but also exceeds the performance of the strongest baseline, PBRS (75.60% at 64 samples). This demonstrates that **GOTTA** adapts using a smaller, yet more informative, subset of data. These findings establish that the quality of the memory curation policy is more critical than the raw quantity of samples for building efficient and robust adaptive systems.

4.3 Ablation

4.3.1 Analysis of Continual Adaptation Performance

In the challenging setting of continual test-time adaptation (PTTA), where models must adapt to a persistent and correlatively sampled non-i.i.d. data stream, our memory category, GOTTA, demonstrates exceptional robustness. The empirical results in Table 2 show that by curating memory based on feature diversity, GOTTA consistently elevates the performance of TTA methods facing these severe, evolving shifts. When instantiated with our proposed policies (FPS, FPSD, or DPP), GOTTA delivers highly competitive accuracy. The strength of this approach is particularly evident when augmenting advanced TTA methods designed for such continual streams. By replacing the native CSTU heuristic in the **RoTTA** , our **FPS** policy improves accuracy from 74.74% to a leading **76.55%**. A similar enhancement is observed with the **NOTE** framework, where **FPS** (72.68%) again surpasses the specialized PBRS policy (70.12%). While simpler strategies like Reservoir Sampling show efficacy with certain backbones (e.g., achieving **21.87%** with TENT), their performance is less consistent across more advanced frameworks. In contrast, our policies demonstrate broad applicability, with **DPP (Ours)** also achieving leading results such as **14.00%** with the Energy backbone and **24.72%** with EATA. This robust performance across diverse adaptation mechanisms provides strong evidence that directly optimizing for representational diversity is a more fundamental principle for sustained adaptation than method-specific heuristics.

4.3.2 Analysis of Episodic Adaptation Performance

In addition to the continual setting, we evaluate GOTTA's performance in an **episodic** adaptation scenario, where the model is reset after each distinct environmental shift. This setup tests the policy's ability to rapidly construct a high-quality, representative memory from scratch. As shown in Table 5, the choice of memory policy is critical for robust adaptation, particularly for advanced TTA backbones. Our diversity-driven policies again demonstrate state-of-the-art performance. For instance, our **FPS** policy elevates **RoTTA**'s accuracy to a leading **75.97%**, surpassing strong class-aware baselines like PBRS (74.48%). Similarly, **FPSD** achieves the highest accuracy with **NOTE** at **74.93%**. We observe that for several TTA methods (e.g., EATA, Energy), the severe episodic shift causes a performance collapse to approximately 24.72% regardless of the memory policy, indicating limitations in the underlying adaptation mechanisms themselves. However, for methods that can effectively leverage a memory buffer, GOTTA's principle of prioritizing feature diversity proves to be a more general and robust strategy for rapid adaptation than heuristics that may rely on stronger assumptions about the data distribution.

4.3.3 IMPACT OF NON-I.I.D. STREAMS ON ADAPTATION

Test-time adaptation is typically evaluated on data streams that are assumed to be approximately i.i.d. However, in realistic deployments this assumption rarely holds: class imbalance and temporal correlations are common. Following prior benchmarks such as NOTE (Gong et al., 2022) and RoTTA (Yuan et al., 2023), we simulate non-i.i.d. streams by sampling class proportions from a Dirichlet distribution. This procedure generates batches that are heavily skewed towards certain classes, creating a challenging setting where naive adaptation can fail.

To control the class distribution of the incoming stream, we sample class proportions from a Dirichlet distribution (Gong et al., 2022; Yuan et al., 2023) with concentration parameter γ where lower values of γ produces more imbalanced (skewed) streams.

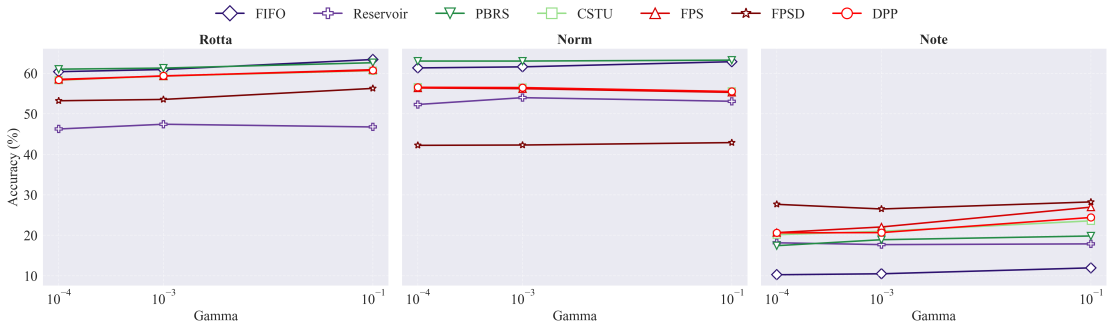


Figure 2: Ablation study on the robustness of TTA methods to data stream correlation on CIFAR-100-C. This figure illustrates the relationship between model accuracy and the hyperparameter γ.

Why Non-i.i.d. Streams Matter. Many TTA methods rely on batch-level statistics (e.g., normalization layers, entropy minimization) and are therefore highly sensitive to class skew. When entire batches are dominated by one class, these updates amplify imbalance and lead to rapid degradation. This makes the role of memory buffers particularly critical: by accumulating samples over time, memory smooths the effective distribution seen during adaptation and mitigates per-batch bias.

Findings. Figure 2, shows that adding even the simplest memory mechanism yields large improvements across methods, with gains ranging from 7% to 53% depending on the baseline. For more details refer to the Appendix Tables 3 and 4. Class-aware memories consistently achieve strong adaptation by enforcing balanced occupancy, though at the cost of storing many redundant samples within each class. Our GOTTA memories reach comparable performance while relying on significantly fewer samples by explicitly promoting intra-class diversity. This demonstrates that it is not the sheer number of stored items that matters, but rather their *diversity*.

5 CONCLUSIONS

In this work, we revisited the role of memory in test-time adaptation and highlighted its often-overlooked impact on stability, efficiency, and robustness. We demonstrated that naive heuristics lead to redundancy and imbalance in the memory, limiting adaptation under realistic, dynamic streams. To overcome this, we introduced GOTTA (Guided Observational Test-Time Adaptation), a principled framework that treats memory as an actively curated coreset rather than a passive buffer. By explicitly enforcing both class balance and intra-class diversity through policies such as Farthest Point Sampling (FPS) and Determinantal Point Processes (DPP), GOTTA preserves a compact yet highly informative memory.

Extensive experiments on CIFAR-C benchmarks and non-i.i.d. stream protocols confirmed that GOTTA consistently improves upon existing methods, achieving higher accuracy with lower memory requirements. Crucially, our results show that performance gains stem from diversity-aware curation rather than buffer size, establishing a new design principle for adaptive systems.

Looking forward, diversity-aware memory opens promising directions for test-time adaptation in more complex settings, including video streams, multi-modal inputs, and real-world online deployments where efficiency and stability are of paramount importance.

REFERENCES

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.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *CVPR*, pp. 295–305, 2022.

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.

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.

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.

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.

Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nat.*, 521(7553):436–444, 2015.

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.

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.

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.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009.

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.

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.

Yongyi Su, Xun Xu, and Kui Jia. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering, 2022. URL https://arxiv.org/abs/2206.02721.

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.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pp. 5987–5995, 2017.

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.

Yang Zou, Zhiding Yu, Xiaofeng Liu, B V K Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, pp. 5982–5991, 2019.