TOWARDS UNBIASED TEST-TIME ADAPTATION FOR VISION–LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Statistics-based test-time adaptation (TTA) provides an efficient way to adapt vision-language models (VLMs) by summarizing test streams into fixed-size class statistics without backpropagation or instance storage. However, there are two limitations in this paradigm. First, persistent class-wise prediction preferences can repeatedly misallocate incoming features, leading to prediction-biased accumulation in the class states. Second, additive feature aggregation can retain class-shared components, leading to classifier homogenization even under correct class allocation. To address these issues, we propose ***U**nbiased **T**est-Time **A**daptation* (**UTA**), which jointly regulates class allocation and classifier construction. Specifically, to mitigate prediction-biased accumulation, a *History-Guided Allocation Debiasing* mechanism is designed to regulate class allocation using historical class responses, thereby reducing allocation errors in the accumulated class states. Furthermore, to alleviate classifier homogenization, we introduce an *Activity-Conditioned Classifier Construction* strategy to reweight accumulated class states according to stream-wide coordinate activity, which limits shared-component dominance and preserves class-specific distinctions. Extensive experiments demonstrate strong adaptation performance across multiple settings while preserving 95% of zero-shot CLIP throughput on large-scale ImageNet-1K. Moreover, the proposed components can serve as lightweight plug-ins for existing statistics-based TTA methods. Our code is available at https://anonymous.4open.science/r/UTA-C574.

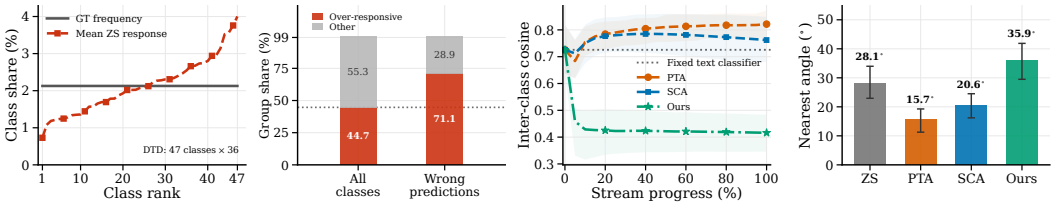
1 INTRODUCTION

Vision-language models (VLMs), such as CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), have shown strong generalization in numerous perception tasks. Since distribution shifts between pretraining data and the target environment are unavoidable, their performance is typically limited in real-world tasks. Recent studies seek to improve downstream performance by adapting VLMs using a small amount of labeled data (Zhou et al., 2022b;a; Khattak et al., 2023a; Chen et al., 2023; Gao et al., 2024). To reduce reliance on costly annotations, test-time adaptation (TTA) has emerged as a promising alternative that enables VLMs to adjust to distribution shifts using only unlabeled test data (Shu et al., 2022; Feng et al., 2023; Huang et al., 2025; Karmanov et al., 2024).

Existing TTA methods for VLMs can be grouped according to whether they require test-time backpropagation. The first paradigm relies on backpropagation-based adaptation to fine-tune the text or visual prompt (Shu et al., 2022; Feng et al., 2023). However, such methods incur substantial computational overhead from backpropagation at test time, limiting their applicability in resource-constrained settings (Zhang et al., 2022). To alleviate this computational burden, prior works have developed backpropagation-free methods, which avoid gradient computation and its associated optimization overhead (Karmanov et al., 2024; Zhang et al., 2024c; Zanella & Ben Ayed, 2024; Zhang et al., 2025). However, removing backpropagation does not necessarily ensure fast inference, as repeated forward passes and auxiliary adaptation operations may still incur substantial costs.

To improve this trade-off, a recent family of backpropagation-free methods summarizes the test stream into fixed-size aggregate statistics (Zhou et al., 2025), which we refer to as *statistics-based TTA*. Under this paradigm, incoming samples update the aggregate statistics online without retaining their individual features or predictions for later retrieval. Existing approaches instantiate this idea through priors from online predictions (Zhou et al., 2025), accumulated feature statistics (Guan

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.



(a) Response mismatch (b) Error destinations (c) Class homogenization (d) Nearest separation

Figure 1: **Prediction bias and classifier homogenization.** (a) Mean zero-shot class probabilities and empirical ground-truth class frequencies. All 10 datasets are provided in Appendix C.1. (b) The fraction of high-response classes and the fraction of erroneous predictions assigned to these classes. (c) Mean pairwise similarity between visual-classifier directions for PTA (Huang et al., 2026) and SCA (Guan et al., 2025), and UTA, with the zero-shot text classifier as a reference. (d) Mean nearest-class angle of the final classifier directions. In (c) and (d), **curves** and **bars** denote dataset averages, while **shaded regions** and **error bars** indicate variation across the ten datasets.

et al., 2025), and confidence-weighted updates of class prototypes (Huang et al., 2026). Despite their different formulations, these methods share a common principle: *prediction-derived class allocation determines how incoming features update the class states, from which an adaptive visual classifier is constructed.* Consequently, the resulting visual classifier captures target-specific visual information during online adaptation, complementing the fixed semantics in text classifier.

Despite their efficiency, statistics-based methods face challenges in class allocation and classifier construction. First, VLM zero-shot responses can exhibit systematic class-wise bias on target data, as illustrated by the response mismatch and error destinations in Fig. 1(a) and (b). When these prediction preferences persist, the induced allocation errors accumulate in class states from which individual sample contributions cannot be selectively recovered and reassigned. Second, correct class allocation alone does not ensure a discriminative visual classifier. Under additive feature aggregation, components shared across classes can persist in the accumulated class states. When shared components dominate the class-dependent contributions, the resulting classifier directions can become less separated. Fig. 1(c) and (d) illustrate this geometric effect in the evaluated classifier.

To address these challenges, we propose ***Unbiased Test-Time Adaptation*** (**UTA**), which jointly regulates class allocation and visual classifier construction. For prediction-biased accumulation, *History-Guided Allocation Debiasing* is introduced to regulate class allocation. Specifically, we maintain an adaptive regulation signal from historical class responses, adjusting the current prediction before feature allocation. As for classifier homogenization, an *Activity-Conditioned Classifier Construction* strategy is further introduced to reweight the accumulated class states before classifier construction. Stream-wide coordinate activity is tracked independently of class allocation and is used to condition the accumulated states, reducing the influence of highly active coordinates while preserving class-specific distinctions. By combining history-guided allocation debiasing with activity-conditioned state reweighting, our method constructs a more discriminative visual classifier that complements the fixed text classifier with target-specific visual information.

Our contributions are summarized as follows: **(1)** We identify and empirically analyze prediction-biased accumulation and classifier homogenization as two limitations of statistics-based TTA. **(2)** We propose ***Unbiased Test-Time Adaptation*** (**UTA**), comprising two lightweight components: History-Guided Allocation Debiasing for regulating prediction-derived class allocation and Activity-Conditioned Classifier Construction for conditioning the visual classifier constructed from accumulated class states. **(3)** Extensive experiments demonstrate a favorable performance–efficiency trade-off, while both components can also serve as lightweight plug-ins for existing TTA methods.

2 Related Work

Vision-Language Models. Vision-language models learn transferable image-text representations from large-scale pretraining, enabling zero-shot visual recognition (Radford et al., 2021; Jia et al., 2021; Zhai et al., 2022; 2023). To adapt these models to downstream tasks, existing studies explore text prompt learning (Zhou et al., 2022b;a; Chen et al., 2023), multimodal and visually informed

prompt learning (Khattak et al., 2023a; Xie et al., 2025), and feature adaptation through lightweight adapters or few-shot caches (Gao et al., 2024; Zhang et al., 2022). Further studies regularize prompt learning to retain pretrained knowledge during adaptation (Yao et al., 2023; Khattak et al., 2023b). These downstream adaptation methods still rely on labeled examples. Our work instead studies adaptation directly from unlabeled test streams, without requiring task-specific annotations.

Test-Time Adaptation for VLMs. Existing TTA methods for VLMs can be broadly divided into backpropagation-based and backpropagation-free approaches. Backpropagation-based approaches optimize learnable parameters, such as prompts and class prototypes, using unlabeled test samples (Shu et al., 2022; Feng et al., 2023; Zhang et al., 2024a; Wang et al., 2026; Song et al., 2026), at the cost of additional gradient computation. Backpropagation-free approaches instead adapt predictions through instance caching (Karmanov et al., 2024; Zhang et al., 2024c), multi-view aggregation (Zanella & Ben Ayed, 2024), or online distribution estimation (Dai & Yang, 2025; Han et al., 2025), without backward passes. Beyond computational efficiency, recent work also investigates prediction and feature debiasing (Wu et al., 2026a;b; Deng et al., 2026; Luo et al., 2026), memory reliability (Zhang et al., 2026), and robustness to varying class coverage and non-i.i.d. test streams (Zanella et al., 2025). We focus on statistics-based test-time adaptation, a paradigm that maintains fixed-size aggregate statistics rather than per-instance memory. Within this paradigm, BCA (Zhou et al., 2025) jointly updates class representations and priors, SCA (Guan et al., 2025) constructs a classifier from accumulated feature statistics, and PTA (Huang et al., 2026) performs confidence-weighted prototype updates. Our focus is on how prediction-derived class allocation and classifier construction from fixed-size aggregate states contribute to prediction-biased accumulation and classifier homogenization, respectively.

3 METHOD

3.1 PRELIMINARIES

Zero-shot CLIP. CLIP (Radford et al., 2021) aligns visual and textual representations through contrastive pretraining. Its visual Φ_v and text encoder Φ_t produce unit-normalized representations in a shared D-dimensional space $\mathbb{R}^D$. For a C-class task, the class names are embedded into prompt templates (e.g., *"a photo of a [class]"*) to generate text classifier $\boldsymbol{e} \in \mathbb{R}^{C\times D}$. Given a test image $\boldsymbol{x}$ in dataset $\mathcal{D}$, let $\boldsymbol{f} = \Phi_v(\boldsymbol{x}) \in \mathbb{R}^{1\times D}$ denote its visual feature. The zero-shot probability of class c is:

$$p_c = \frac{exp\left(cos(\boldsymbol{f}, \boldsymbol{e}_c)/\tau\right)}{\sum_{j=1}^{C} exp\left(cos(\boldsymbol{f}, \boldsymbol{e}_j)/\tau\right)}, \tag{1}$$

where τ is the temperature and $cos(\cdot,\cdot)$ is cosine similarity. The prediction is $\hat{y} = \arg\max_c p_c$.

Statistics-based test-time adaptation. Statistics-based TTA reduces adaptation overhead by maintaining aggregate statistics (Zhou et al., 2025; Guan et al., 2025; Huang et al., 2026). It maintains fixed-size aggregate statistics without retaining individual test samples, features, or predictions for later retrieval. Their class-wise accumulation component follows an allocation-and-aggregation principle: prediction-derived weights determine how each incoming feature updates the class states used for classifier construction. For this class-wise accumulation component, let $\boldsymbol{P}_t \in \mathbb{R}^{C\times D}$ denote the accumulated class states at test step t. A generic state update can be written as:

$$\boldsymbol{P}_t = \boldsymbol{\Lambda}_t \boldsymbol{P}_{t-1} + \boldsymbol{A}_t \boldsymbol{f}_t. \tag{2}$$

where $\boldsymbol{f}_t$ is the current visual feature, $\boldsymbol{A}_t \in \mathbb{R}^{C\times 1}$ is the class-allocation vector, and $\boldsymbol{\Lambda}_t \in \mathbb{R}^{C\times C}$ controls how the previous class states are retained (e.g., Exponential Moving Average, accumulation). Thus, $\boldsymbol{A}_t \boldsymbol{f}_t$ allocates the current feature across class states, while $\boldsymbol{\Lambda}_t \boldsymbol{P}_{t-1}$ carries forward the retained states. Different methods instantiate these terms differently (e.g., row normalization) and may maintain additional statistics to construct the visual classifier from the accumulated class states.

3.2 UNBIASED TEST-TIME ADAPTATION

Overview. To address prediction-biased accumulation and classifier homogenization in statistics-based TTA, UTA comprises two complementary components, as illustrated in Fig. 2. For class allocation, a class-wise preference offset is derived from historical class responses to regulate class allocation. For discriminative visual classifier construction, the accumulated class states are conditioned on stream-wide coordinate activity to construct the visual classifier.

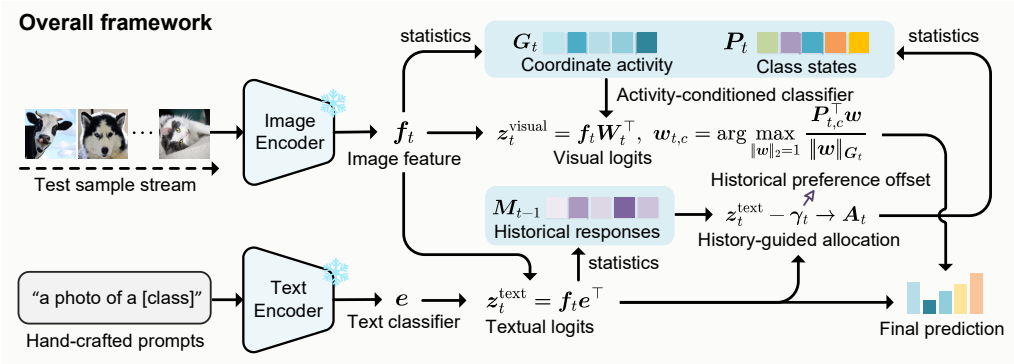


Figure 2: Overview of **UTA**. Class allocation is regulated by a historical preference offset, while the visual classifier is constructed from activity-conditioned class states. Visual logits are fused with original textual logits, complementing fixed textual semantics with target-specific visual information.

History-guided Allocation Debiasing (HAD). The class-allocation vector $\boldsymbol{A}_t$ in Eq. 2 determines how the current visual feature updates the class states. Once incorporated, an individual contribution cannot be selectively reassigned from the retained aggregate states alone. Persistent prediction preferences can therefore induce allocation errors that distort the accumulated class states.

To examine how class-wise prediction preferences affect allocation, for a test sample x, let $z_c(x)$ denote the zero-shot logit for class c. We consider the idealized decomposition $z_c(x) = \log p_c^\star(x) + \beta_c$, up to a class-independent additive constant, where $p_c^\star(x)$ is a bias-free reference probability and β_c is a sample-independent offset for class c. For classes c and j, the score difference is $z_c(x) - z_j(x) = \log[p_c^\star(x)/p_j^\star(x)] + (\beta_c - \beta_j)$. Thus, $\beta_c - \beta_j$ introduces a sample-independent score advantage between the two classes. This motivates class-wise logit correction before feature aggregation.

Although the latent offsets are unobservable, their effect is reflected in persistent class-wise preferences in the zero-shot probabilities. Motivated by this observation, we maintain a fixed-size adaptive state $\boldsymbol{M} \in \mathbb{R}^C$ to summarize class-wise VLM responses over the test stream. Conceptually, it can be accumulated as $\boldsymbol{M}_t \leftarrow \boldsymbol{M}_{t-1} + \boldsymbol{p}_t$, where $\boldsymbol{p}_t$ denotes the VLM probability vector. At test step t, $\boldsymbol{M}_{t-1}$ summarizes the responses accumulated from preceding samples, from which we compute:

$$\boldsymbol{\ell}_t = \log(1 + \boldsymbol{M}_{t-1}), \tag{3}$$

Let $M_{t,c}$ denote the historical state associated with class c. For classes c and j, $\ell_{t,c} - \ell_{t,j} = \log[(M_{t-1,c} + 1)/(M_{t-1,j} + 1)]$, which expresses their relative historical response as an additive log difference. Then, the relative historical-response imbalance can be summarized by:

$$\boldsymbol{\delta}_t = \operatorname{arsinh}\left(\frac{\boldsymbol{\ell}_t - \operatorname{mean}(\boldsymbol{\ell}_t)}{\operatorname{std}(\boldsymbol{\ell}_t)}\right), \tag{4}$$

where the mean and population standard deviation are computed across classes, and the remaining operations are element-wise. Centering and standardization express historical responses relative to their cross-class location and scale, while arsinh compresses large standardized deviations. The resulting response imbalance is applied to the textual zero-shot logits $\boldsymbol{z}_t^{\text{text}} = \boldsymbol{f}_t \boldsymbol{e}^\top$ by:

$$\boldsymbol{\gamma}_t = \omega_t \boldsymbol{\delta}_t, \qquad \boldsymbol{q}_t = \operatorname{softmax}(\boldsymbol{z}_t^{\text{text}} - \boldsymbol{\gamma}_t). \tag{5}$$

where $\omega_t = t/(t + C)$ controls the strength of historical regulation and gradually increases its influence as the stream progresses. The corrected response $\boldsymbol{q}_t$ determines the current class allocation $\boldsymbol{A}_t = \mathcal{S}_k(\boldsymbol{q}_t)$, where $\mathcal{S}_k$ retains and renormalizes the top k entries. We use $k = 1$, corresponding to hard allocation for all experiments. After $\boldsymbol{q}_t$ is obtained, it is also incorporated into the historical state for subsequent samples: $\boldsymbol{M}_t = \boldsymbol{M}_{t-1} + \boldsymbol{q}_t$. This recursive update yields an adaptive historical state in which corrected responses from preceding samples regulate subsequent class allocation.

For a nonconstant historical state and $\omega_t > 0$, the transformations preserve class ordering: $M_{t-1,c} > M_{t-1,j}$ implies $\gamma_{t,c} > \gamma_{t,j}$. For $\gamma_{t,c} > \gamma_{t,j}$, class c has a strictly higher corrected logit than class j only if $z_{t,c}^{\text{text}} - z_{t,j}^{\text{text}} > \gamma_{t,c} - \gamma_{t,j}$. Thus, a class with a stronger historical response must overcome a larger score difference before receiving another feature.

For an idealized analysis of hard-allocation error, additionally assume $p_c^\star(x) = \Pr_{\mathcal{D}}(y = c \mid x)$ and consider a fixed correction γ satisfying

$$0 \le \gamma_c - \gamma_j \le \beta_c - \beta_j, \qquad \forall c, j \text{ such that } \beta_c \ge \beta_j. \tag{6}$$

This condition aligns the correction with the offset ordering without over-correction; $\gamma = \alpha\beta$, with $0 \le \alpha \le 1$, is a sufficient special case. Let $\hat{y}(x) = \arg\max_c z_c(x)$ and $\hat{y}_\gamma(x) = \arg\max_c [z_c(x) - \gamma_c]$, using the same fixed class ordering to break ties. Under Eq. (6), the corrected selection has a target posterior no smaller than that of the uncorrected selection. Consequently, the hard-allocation error $\epsilon_{\text{alloc}}(\hat{y}) = \Pr_{(x,y)\sim\mathcal{D}}[\hat{y}(x) \neq y]$ for any target sample $(x, y) \in \mathcal{D}$ satisfies:

$$\epsilon_{\text{alloc}}(\hat{y}_\gamma) \le \epsilon_{\text{alloc}}(\hat{y}). \tag{7}$$

Proposition 1 in Appendix A.1 provides the formal proof. The sufficient condition above is not guaranteed for the adaptive correction γ_t. This result motivates using historical responses as a control signal for class allocation rather than as a direct estimator of the latent offset β or the target prior. Empirically, HAD improves average hard-allocation accuracy across the ten datasets in Fig. 3.

Activity-Conditioned Classifier Construction (ACC). Correct class allocation alone does not ensure well-separated classifier directions. To examine classifier construction without allocation errors, consider an oracle that assigns each feature $\boldsymbol{f}_i$ to its ground-truth class y_i. Let $\boldsymbol{\mu}_t = t^{-1}\sum_{i=1}^{t} \boldsymbol{f}_i$ denote the observed stream mean after $t \ge 1$ samples. With text-classifier $\boldsymbol{e}$ initialization, for any class c, the oracle class state $\boldsymbol{s}_t^{\text{GT}}$ after t updates can be decomposed as:

$$\boldsymbol{s}_{t,c}^{\text{GT}} = \boldsymbol{e}_c + \sum_{i \le t : y_i = c} \boldsymbol{f}_i = \boldsymbol{e}_c + N_{t,c}\boldsymbol{\mu}_t + \sum_{i \le t : y_i = c} (\boldsymbol{f}_i - \boldsymbol{\mu}_t), \tag{8}$$

where $N_{t,c} = \sum_{i=1}^{t} \mathbf{1}\{y_i = c\}$. This decomposition separates each class state into a fixed text classifier, a count-scaled stream-wide component $N_{t,c}\mu_t$, and a class-dependent residual. After row normalization, if the shared component dominates the class-specific terms, different class states are driven toward the same stream-wide direction, revealing a source of classifier homogenization even under oracle allocation. This motivates accounting for the anisotropic structure of the target stream when constructing classifier directions, as coordinates that remain highly active throughout the stream can disproportionately influence the accumulated class states.

To obtain a class-agnostic reference for such stream-wide activity, we track the accumulated squared activity of each feature coordinate. The coordinate-activity vector can be defined by $\boldsymbol{G}_t = \mathbf{1} + \sum_{i=1}^{t} \boldsymbol{f}_i \odot \boldsymbol{f}_i$, where $\odot$ denotes element-wise multiplication. The unit baseline keeps every entry positive, while squaring prevents sign cancellation and accumulates the magnitude of activity on each coordinate. The positive activity vector $\boldsymbol{G}_t$ defines an activity-weighted norm for classifier construction. Specifically, for any direction $\boldsymbol{w} \in \mathbb{R}^D$, this squared norm is

$$\|\boldsymbol{w}\|_{\boldsymbol{G}_t}^2 = \sum_{d=1}^{D} G_{t,d} w_d^2 = \|\boldsymbol{w}\|_2^2 + \sum_{i=1}^{t} \|\boldsymbol{w} \odot \boldsymbol{f}_i\|_2^2. \tag{9}$$

This norm penalizes classifier directions that place more weight on coordinates with high accumulated squared activity. For any nonzero class-state vector $\boldsymbol{s} \in \mathbb{R}^D$, we select the unit direction that maximizes its alignment with $\boldsymbol{s}$ relative to the activity-weighted norm:

$$\boldsymbol{w}^\star = \arg\max_{\|\boldsymbol{w}\|_2 = 1} \frac{\boldsymbol{s}^\top \boldsymbol{w}}{\|\boldsymbol{w}\|_{\boldsymbol{G}_t}} = \frac{\boldsymbol{s} \oslash \boldsymbol{G}_t}{\|\boldsymbol{s} \oslash \boldsymbol{G}_t\|_2}. \tag{10}$$

where $\oslash$ denotes element-wise division. The derivation is provided in Appendix A.2. By the Cauchy–Schwarz inequality, the maximum is attained at a positive multiple of $s \oslash \boldsymbol{G}_t$. Coordinates with larger accumulated activity receive smaller scaling factors, while their final contributions remain modulated by the corresponding class-state values. This criterion specifies how a fixed class-state vector is converted into a classifier direction. Using the allocation $\boldsymbol{A}_t = \mathcal{S}_k(\boldsymbol{q}_t)$, we initialize the class states with the text classifier $\boldsymbol{P}_0 = \boldsymbol{e}$, and set $\boldsymbol{G}_0 = \mathbf{1}$. The two states are updated as

$$\boldsymbol{P}_t = \boldsymbol{P}_{t-1} + \boldsymbol{A}_t \boldsymbol{f}_t, \qquad \boldsymbol{G}_t = \boldsymbol{G}_{t-1} + \boldsymbol{f}_t \odot \boldsymbol{f}_t, \tag{11}$$

Here, $\boldsymbol{P}_t$ stores the accumulated class states under corrected class allocation, whereas $\boldsymbol{G}_t$ tracks coordinate-wise squared activity without using class-allocation weights. Since the update of $\boldsymbol{G}_t$ does

Table 1: Results on cross-dataset generalization using ViT-B/16 (Dosovitskiy et al., 2021) as the backbone. For a fair comparison, we adopt the prompt templates described in the original CLIP (Radford et al., 2021), where † denotes results reproduced using the officially released code.

Method	Flowers	DTD	Pets	Cars	UCF	Cal.	Food	SUN	Air.	Euro.	Avg.
CLIP† (Radford et al., 2021)	70.73	44.09	89.07	66.21	67.54	93.91	85.90	66.31	24.72	48.21	65.67
Backpropagation-Based Test-Time Adaptation											
TPT (Shu et al., 2022)	68.98	47.75	87.79	68.50	68.04	94.16	84.67	65.50	24.78	42.44	65.26
DiffTPT (Feng et al., 2023)	70.10	47.00	88.22	67.01	62.67	92.49	87.23	65.74	25.60	43.13	64.92
HisTPT (Zhang et al., 2024b)	71.20	48.90	89.10	69.20	70.10	94.50	89.30	67.20	26.90	49.70	67.61
GS-Bias (Huang et al., 2025)	71.94	46.10	90.38	67.33	67.59	94.60	86.09	67.40	26.49	52.42	67.03
DLAE† (Wu et al., 2026a)	73.37	49.82	90.87	69.69	73.75	94.65	87.17	68.70	28.23	61.52	69.78
Backpropagation-Free Test-Time Adaptation											
MTA† (Zanella & Ben Ayed, 2024)	71.58	45.09	89.37	68.37	69.50	94.44	**87.06**	67.78	26.91	47.49	66.76
TDA† (Karmanov et al., 2024)	71.21	44.86	89.67	67.43	70.55	94.08	86.14	67.84	25.32	61.75	67.89
TCA† (Wang et al., 2025)	73.08	44.03	89.07	65.54	69.28	93.87	84.61	67.03	24.51	**66.63**	67.77
ADAPT† (Zhang et al., 2025)	73.04	47.04	91.36	69.01	70.63	93.23	85.56	69.08	24.93	64.41	68.83
BCA† (Zhou et al., 2025)	70.44	44.21	88.85	66.32	67.88	93.96	85.92	66.50	25.20	49.57	65.89
SCA† (Guan et al., 2025)	74.26	48.40	90.11	67.72	71.11	93.75	86.06	68.96	25.14	49.58	67.51
PTA† (Huang et al., 2026)	**74.91**	47.22	91.33	68.65	73.22	**94.65**	86.34	69.12	25.77	61.36	69.26
UTA	74.42	**49.23**	**92.07**	**70.51**	**75.68**	94.24	86.79	**70.93**	**27.03**	65.64	**70.66**

not use $\boldsymbol{A}_t$, allocation errors do not directly affect the coordinate-activity reference. Let $\boldsymbol{P}_{t,c}$ denote the accumulated state of class c, applying $\boldsymbol{s} = \boldsymbol{P}_{t,c}$ to Eq. 10 gives the classifier direction $\boldsymbol{w}_{t,c}$:

$$\boldsymbol{w}_{t,c} = \frac{\boldsymbol{P}_{t,c} \oslash \boldsymbol{G}_t}{\|\boldsymbol{P}_{t,c} \oslash \boldsymbol{G}_t\|_2}. \tag{12}$$

Stacking the normalized class directions $w_{t,c}$ row-wise yields the conditioned visual classifier $\boldsymbol{W}_t = [\boldsymbol{w}_{t,1}, \ldots, \boldsymbol{w}_{t,C}]^\top \in \mathbb{R}^{C \times D}$. Thus, ACC determines how the accumulated class states form the visual classifier, complementing HAD's correction of the logits used for class allocation.

Prediction. The ACC-conditioned classifier produces the visual logits $\boldsymbol{z}_t^{\text{visual}} = \boldsymbol{f}_t \boldsymbol{W}_t^\top$. Finally, the visual logits are then fused with the original textual zero-shot logits $\boldsymbol{z}_t^{\text{text}}$ by:

$$\boldsymbol{z}_t^{\text{UTA}} = \boldsymbol{z}_t^{\text{text}} + \boldsymbol{z}_t^{\text{visual}}. \tag{13}$$

This fusion combines the fixed original textual logits with target-specific visual information. The final prediction is $\hat{y}_t = \arg\max_c z_{t,c}^{\text{UTA}}$. Appendix B summarizes the online procedure.

4 Experiments

Datasets. We evaluate UTA on 15 image classification datasets and 4 corrupted point cloud benchmarks, assessing its effectiveness across both image and point cloud modalities.

For image classification, we consider ImageNet-1K (Deng et al., 2009; Russakovsky et al., 2015), 10 cross-dataset benchmarks, and 4 out-of-distribution (OOD) benchmarks. The cross-dataset evaluation covers diverse recognition tasks using Caltech101 (Fei-Fei et al., 2004), OxfordPets (Parkhi et al., 2012), Flowers102 (Nilsback & Zisserman, 2008), StanfordCars (Krause et al., 2013), FGVC-Aircraft (Maji et al., 2013), Food101 (Bossard et al., 2014), SUN397 (Xiao et al., 2010), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), and UCF101 (Soomro et al., 2012). To assess robustness to distribution shifts, we additionally evaluate on ImageNet-V2 (Recht et al., 2019), ImageNet-Sketch (Wang et al., 2019), ImageNet-A (Hendrycks et al., 2021b), and ImageNet-R (Hendrycks et al., 2021a), with the original ImageNet-1K validation set serving as the in-distribution reference.

For point cloud evaluation, we follow (Huang et al., 2026), adopt the protocol of Point-Cache (Sun et al., 2025) and use ModelNet-C (Ren et al., 2022), together with corrupted versions of three ScanObjectNN variants (Uy et al., 2019). The corruption protocol comprises seven operations: rotation, scaling, jittering, global and local outlier injection, and global and local point removal.

Table 2: Results on ImageNet and OOD generalization using ViT-B/16 as the backbone. Average and OOD Average denote the mean accuracy over 5 datasets and the 4 OOD datasets, respectively.

Method	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-S	Average	OOD Average
CLIP (Radford et al., 2021)	66.73	47.87	60.86	73.98	46.09	59.11	57.20
Backpropagation-Based Test-Time Adaptation							
TPT (Shu et al., 2022)	68.98	54.77	63.45	77.06	47.94	62.44	60.81
DiffTPT (Feng et al., 2023)	70.30	55.68	65.10	75.00	46.80	62.58	60.65
DPE (Zhang et al., 2024a)	71.91	59.63	65.44	80.40	52.26	65.93	64.43
GS-Bias (Huang et al., 2025)	70.57	56.61	64.62	80.49	50.33	64.52	63.01
DLAE† (Wu et al., 2026a)	71.34	63.25	65.20	79.94	52.32	66.41	65.18
Backpropagation-Free Test-Time Adaptation							
MTA (Zanella & Ben Ayed, 2024)	70.08	58.06	64.24	78.33	49.61	64.06	62.56
TDA (Karmanov et al., 2024)	69.51	60.11	64.67	80.24	50.54	65.01	63.89
ADAPT (Zhang et al., 2025)	70.55	60.87	64.44	**80.85**	51.49	65.64	64.41
BCA† (Zhou et al., 2025)	68.87	59.91	64.30	80.44	49.80	64.66	63.61
SCA† (Guan et al., 2025)	70.20	**61.01**	63.92	80.80	51.15	65.42	64.22
PTA† (Huang et al., 2026)	70.20	60.53	64.63	80.14	50.54	65.21	63.96
UTA	**71.39**	60.96	**65.12**	80.19	**52.98**	**66.13**	**64.81**

Implementation details. For fair comparison, we follow the experimental settings of (Shu et al., 2022; Karmanov et al., 2024). We use CLIP (Radford et al., 2021) for image recognition, with ViT-B/16 (Dosovitskiy et al., 2021) as the visual encoder. The batch size is set to 1 to satisfy online processing requirements. For textual prompts, we follow prior work (Huang et al., 2026) and adopt the ensemble prompting strategy described in the original CLIP, without generating prompts with a large language model (LLM) for a fair comparison. The top-1 accuracy is used to measure classification performance. All experiments were conducted on a single NVIDIA RTX 5090 GPU.

4.1 Comparison with state-of-the-art methods

Baselines. To evaluate the effectiveness of UTA, we compare it with zero-shot CLIP (Radford et al., 2021) and 13 TTA methods, divided into two categories. The first includes six backpropagation-based methods: TPT (Shu et al., 2022), DiffTPT (Feng et al., 2023), DPE (Zhang et al., 2024a), HisTPT (Zhang et al., 2024b), GS-Bias (Huang et al., 2025), and DLAE (Wu et al., 2026a). The second includes seven backpropagation-free methods: MTA (Zanella & Ben Ayed, 2024), TDA (Karmanov et al., 2024), TCA (Wang et al., 2025), ADAPT (Zhang et al., 2025), BCA (Zhou et al., 2025), SCA (Guan et al., 2025), and PTA (Huang et al., 2026). Among them, BCA, SCA, and PTA are statistics-based methods and offer the most directly comparable baselines to our method.

Cross-dataset generalization. As shown in Table 1, UTA achieves the highest average accuracy of **70.66%** among all compared methods, improving over zero-shot CLIP by **4.99**%. Compared with the statistics-based methods BCA, SCA, and PTA, the average gains are **4.77**%, **3.15**%, and **1.40**% respectively. Among the backpropagation-free methods, UTA achieves the best accuracy on six of the ten datasets. These results demonstrate the effectiveness of UTA across diverse recognition tasks and its improvement over existing statistics-based adaptation methods.

Out-of-distribution generalization. We further evaluate UTA on ImageNet and four OOD benchmarks. As reported in Table 2, UTA achieves **71.39**% accuracy on ImageNet, a **64.81%** OOD average, and a **66.13%** average across all five datasets. Among the backpropagation-free methods, UTA obtains the highest overall and OOD averages. These results show that its performance gains extend beyond cross-dataset recognition to shifted test distributions.

Robust point cloud recognition. To examine whether our UTA transfers beyond image recognition, we evaluate ULIP (Xue et al., 2023) on ModelNet-C (Ren et al., 2022) and three ScanObjectNN-C (Uy et al., 2019) variants. Following PTA (Huang et al., 2026), the evaluation covers seven corruption types at severity level 2. As shown in Table 3, our UTA achieves the highest average accuracy on all four benchmarks. These results demonstrate the transferability of our UTA.

Table 3: Results of recognition accuracy on ModelNet-C (Ren et al., 2022) and 3 corrupted variants of ScanObjectNN-C (Uy et al., 2019), following the protocol of PTA (Huang et al., 2026).

Method	Add Global	Add Local	Drop Global	Drop Local	Rotate	Scale	Jitter	Avg.
ModelNet-C								
ULIP† (Xue et al., 2023)	33.83	43.76	54.70	50.78	55.27	50.65	43.99	47.57
Point-Cache (Sun et al., 2025)	46.15	47.85	59.16	56.00	61.47	55.35	49.92	53.70
PTA (Huang et al., 2026)	48.78	51.17	61.54	56.69	61.42	56.44	49.88	55.13
UTA	**50.18**	**51.53**	**62.29**	**58.00**	**61.56**	**56.86**	49.35	**55.68**
ScanObjectNN-C (OBJ-ONLY)								
ULIP† (Xue et al., 2023)	31.50	34.94	51.12	39.19	48.59	44.18	36.55	40.87
Point-Cache (Sun et al., 2025)	32.01	38.04	54.56	45.27	50.95	45.96	39.24	43.72
PTA (Huang et al., 2026)	34.76	39.76	53.53	45.78	**51.81**	46.47	**39.59**	44.53
UTA	**35.92**	**40.10**	**55.02**	**47.68**	50.89	**47.73**	38.27	**45.09**
ScanObjectNN-C (OBJ-BG)								
ULIP† (Xue et al., 2023)	26.45	27.77	45.78	34.42	41.02	39.01	30.06	34.93
Point-Cache (Sun et al., 2025)	28.23	30.12	48.71	40.45	43.55	40.28	**34.42**	37.97
PTA (Huang et al., 2026)	**32.36**	30.98	**52.15**	42.00	48.53	46.64	32.19	40.69
UTA	32.07	**33.22**	**52.15**	**43.66**	**49.34**	**47.05**	32.76	**41.46**
ScanObjectNN-C (hardest)								
ULIP† (Xue et al., 2023)	19.21	18.52	30.92	23.99	27.19	26.49	21.46	23.97
Point-Cache (Sun et al., 2025)	23.46	22.69	34.70	31.75	33.00	28.28	**25.05**	28.42
PTA (Huang et al., 2026)	24.18	22.07	38.41	**32.51**	34.21	**30.78**	23.63	29.40
UTA	**27.17**	**25.49**	**38.43**	32.47	**34.92**	30.62	24.46	**30.51**

4.2 Model Analysis

Plug-in effectiveness. We further evaluate the proposed UTA as plug-ins for six existing TTA methods on cross-dataset generalization benchmarks. In Table 4, both HAD and ACC are incorporated into the statistics-based methods (left), whereas HAD is applied to the remaining methods (right). All six baselines benefit from these additions, with an average accuracy gain of **1.61**%. These results indicate that UTA complements different adaptation strategies. The per-dataset results are provided in Appendix C.5.

Table 4: Effectiveness of our UTA components as plug-ins for existing methods (%).

Method	Accuracy	Method	Accuracy
BCA	65.89	TCA	67.77
+UTA	**68.26**	**+UTA**	**69.55**
SCA	67.51	DPE	68.28
+UTA	**69.88**	**+UTA**	**69.41**
PTA	69.26	TDA	67.89
+UTA	**70.40**	**+UTA**	**68.74**

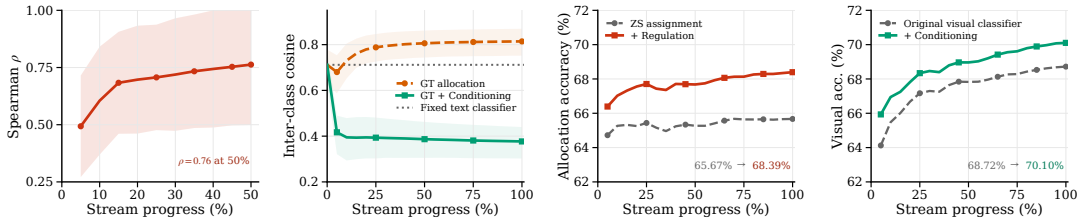
Component ablation. We disentangle the contributions of HAD and ACC on the ten cross-dataset benchmarks. The common baseline combines the zero-shot branch with a $\boldsymbol{P}_t$-based visual classifier using standard row normalization. HAD and ACC provide complementary improvements on class allocation and classifier construction, with their combination achieving the best performance. The per-dataset results are provided in Appendix C.4.

Table 5: Component ablations (%).

Visual classifier	HAD	ACC	Acc.
			65.67
✓			67.11
✓	✓		69.96
✓		✓	67.49
✓	✓	✓	**70.66**

Historical-response analysis. To assess whether raw zero-shot responses provide a useful signal for class allocation, we examine their association with subsequent prediction bias. For each dataset, we compute each class's mean zero-shot probability over a growing prefix of the first half and its prediction excess in the fixed second half, defined as the mean probability minus the ground-truth class frequency in that half. We calculate Spearman's ρ between the rankings of these two quantities across classes, then average the correlations over ten datasets. The average reaches $\rho = 0.76$ at the stream midpoint (Fig. 3(a)), suggesting that classes with stronger historical responses tend to receive greater excess probability later. To isolate HAD from the visual classifier, we additionally evaluate a zero-shot-only variant using the same corrected-history update as the full method. This improves average allocation accuracy from 65.67% to 68.39% (Fig. 3(c)), supporting the use of corrected historical responses for subsequent allocation regulation.

Classifier-construction analysis. To examine the need for classifier regulation beyond allocation correction, we construct visual classifier with and without ACC from the same class states accumulated



(a) History–future link (b) GT homogenization (c) Allocation accuracy (d) Visual classifier acc.

Figure 3: **Mechanism analysis on cross-dataset benchmarks.** (a) Class-wise Spearman correlation between historical raw zero-shot responses and subsequent prediction excess. (b) Mean inter-class cosine similarity of the original and ACC-conditioned classifier under GT allocation. (c) Cumulative allocation accuracy of zero-shot and corrected predictions. (d) Effect of ACC on cumulative visual-only accuracy. **Curves** show equally weighted averages across datasets. **Shading** indicates one standard deviation in (a) and the interquartile range in (b), both across datasets. Dataset-wise summary statistics for the mechanism diagnostics are provided in Appendix C.2.

under ground-truth allocation. The mean inter-class cosine similarity reaches 0.81 without ACC, compared with 0.38 with ACC (Fig. 3(b)). This supports our analysis that correct allocation alone does not ensure well-separated classifier directions. Under identical predicted assignments, ACC also improves average visual-only accuracy from 68.72% to 70.10% (Fig. 3(d)). These results support ACC's use of stream-wide coordinate activity to rescale accumulated class states.

Efficiency analysis. We compare inference throughput and GPU memory usage on ImageNet-1K in Table 6. UTA processes **181.1** FPS, retaining approximately **95**% of zero-shot CLIP's throughput. Its memory usage is **358.4** MiB, only **2.0** MiB higher than that of CLIP. Compared with PTA, UTA improves accuracy from 70.20% to 71.39% and increases throughput from 164.8 to 181.1 images per second, with the same reported memory usage. It also achieves approximately **17**× the throughput of DLAE while obtaining slightly higher ImageNet accuracy. These results demonstrate that UTA improves adaptation performance while maintaining near-zero-shot efficiency.

Table 6: Efficiency analysis on ImageNet-1K.

Method	Acc. (%) ↑	FPS (Image/Second) ↑	Memory ↓
CLIP	66.73	190.6 (100%)	356.4 MiB
Backpropagation-Based Test-Time Adaptation			
TPT	68.98	2.9 (2%)	16339.0 MiB
DPE	71.91	6.6 (3%)	5700.4 MiB
GS-Bias	70.57	14.6 (8%)	702.2 MiB
DLAE	71.34	10.7 (6%)	4208.0 MiB
Backpropagation-Free Test-Time Adaptation			
MTA	70.08	16.1 (8%)	1008.3 MiB
BoostAdapter	71.08	3.2 (2%)	707.6 MiB
ADAPT	70.55	4.2 (2%)	569.4 MiB
BCA	68.87	170.8 (90%)	359.7 MiB
SCA	70.20	108.7 (57%)	363.0 MiB
PTA	70.20	164.8 (86%)	**358.4 MiB**
UTA	**71.39**	**181.1** (95%)	**358.4 MiB**

More analysis. Due to space limitations, additional analyses of class-wise response mismatch C.1, dataset-wise mechanism diagnostics C.2, prompt robustness C.3, robustness to test-stream ordering and non-IID distributions (C.7- C.8), and different VLM backbones C.6 are provided in the appendix.

5 Conclusion

In this paper, we proposed **UTA**, a lightweight statistics-based test-time adaptation method for vision–language models. Specifically, History-Guided Allocation Debiasing (HAD) is introduced to mitigate prediction-biased accumulation by maintaining an adaptive history of corrected class responses and regulating subsequent class allocation. To alleviate classifier homogenization, Activity-Conditioned Classifier Construction (ACC) is further designed to reweight accumulated class states according to stream-wide coordinate activity before classifier construction. By combining these two components, a more discriminative visual classifier is constructed from the test stream, complementing the fixed text classifier with target-specific visual information. Extensive experiments across diverse benchmarks validate the effectiveness and efficiency of our method, while further demonstrating the transferability of the proposed components to existing TTA methods.

AI USE STATEMENT

In this work, we used generative AI tools to support research execution and analysis, including feedback on the design and interpretation of the dataset-wise mechanism diagnostics, plug-in effectiveness, and efficiency analyses, as well as assistance with reproducing and evaluating existing methods used in these experiments. Generative AI tools were also used to organize and review the accompanying code linked in the abstract, assist in checking its consistency with the reported results and experimental settings, and identify any potential disclosure of author-identifying or confidential information. We further used generative AI tools to help check mathematical formulations and theoretical proofs. Additionally, generative AI tools were used to aid and polish the manuscript, including language editing and presentation. All AI-assisted text, code, analyses, and mathematical arguments were reviewed and revised by the authors. The authors manually verified the reproduced methods, accompanying code, reported results, experimental configurations, and anonymization before submission, and take full responsibility for the final content of this work.

REFERENCES

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *Computer Vision–ECCV 2014*, 2014. URL https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/.

Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. PLOT: Prompt learning with optimal transport for vision-language models. In *International Conference on Learning Representations*, 2023.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. URL https://www.robots.ox.ac.uk/~vgg/publications/2014/Cimpoi14/.

Qiyuan Dai and Sibei Yang. Free on the fly: Enhancing flexibility in test-time adaptation with online EM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9538–9548, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Ruxi Deng, Wenxuan Bao, Tianxin Wei, and Jingrui He. Panda: Test-time adaptation with negative data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3551–3559, 2026. doi: 10.1609/aaai.v40i5.37353.

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 *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2004. URL https://cs.stanford.edu/groups/vision/documents/Fei-FeiFergusPerona_GMBV04.pdf.

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

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. CLIP-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132:5813–5831, 2024.

Zenghao Guan, Yucan Zhou, Wu Liu, and Xiaoyan Gu. Statistics caching test-time adaptation for vision-language models. In *Advances in Neural Information Processing Systems*, 2025.

Zongbo Han, Jialong Yang, Guangyu Wang, Junfan Li, Qianli Xu, Mike Zheng Shou, and Changqing Zhang. DOTA: Distributional test-time adaptation of vision-language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/d24189d251aa453b6d2eb3421ec48bcc-Abstract-Conference.html.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. URL https://github.com/phelber/EuroSAT.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021a. URL https://openaccess.thecvf.com/content/ICCV2021/html/Hendrycks_The_Many_Faces_of_Robustness_A_Critical_Analysis_of_Out-of-Distribution_ICCV_2021_paper.html.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271, 2021b. URL https://openaccess.thecvf.com/content/CVPR2021/html/Hendrycks_Natural_Adversarial_Examples_CVPR_2021_paper.html.

Zhaohong Huang, Yuxin Zhang, Jingjing Xie, Fei Chao, and Rongrong Ji. GS-bias: Global-spatial bias learner for single-image test-time adaptation of vision-language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26162–26174, 2025.

Zhaohong Huang, Yuxin Zhang, Wenjing Liu, Fei Chao, and Rongrong Ji. Prototype-based test-time adaptation of vision-language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 4904–4916, 2021.

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

Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. MaPLe: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19113–19122, 2023a.

Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15190–15200, 2023b.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013. URL https://openaccess.thecvf.com/content_iccv_workshops_2013/W19/html/Krause_3D_Object_Representations_2013_ICCV_paper.html.

Yihong Luo, Wenwu He, Dong Liang, Yihang Zhou, and Zhuo-Xu Cui. D^2O: A dual debiasing operator for training-free test-time adaptation of vision–language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. URL https://arxiv.org/abs/1306.5151.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. URL https://www.robots.ox.ac.uk/~vgg/publications/2008/Nilsback08/.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. URL https://www.robots.ox.ac.uk/~vgg/publications/2012/parkhi12a/.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 5389–5400, 2019. URL https://proceedings.mlr.press/v97/recht19a.html.

Jiawei Ren, Liang Pan, and Ziwei Liu. Benchmarking and analyzing point cloud classification under corruptions. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14274–14289, 2022.

Fei Song, Yi Li, Rui Wang, Jiahuan Zhou, Changwen Zheng, and Jiangmeng Li. Doubly debiased test-time prompt tuning for vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 9069–9078, 2026. doi: 10.1609/aaai.v40i11.37863.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. Technical Report CRCV-TR-12-01, Center for Research in Computer Vision, University of Central Florida, 2012. URL https://www.crcv.ucf.edu/papers/UCF101_CRCV-TR-12-01.pdf.

Hongyu Sun, Qiuhong Ke, Ming Cheng, Yongcai Wang, Deying Li, Chenhui Gou, and Jianfei Cai. Point-cache: Test-time dynamic and hierarchical cache for robust and generalizable point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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

Haohan Wang, Songwei Ge, Eric P. Xing, and Zachary C. Lipton. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper/2019/hash/3eefceb8087e964f89c2d59e8a249915-Abstract.html.

Xucong Wang, Zhe Zhao, Zekun Wang, Xiaofeng Cao, Xu Wang, Di Wu, Pengkun Wang, and Yang Wang. Long-tailed test-time adaptation for vision-language models. In *International Conference on Learning Representations*, 2026.

Zixin Wang, Dong Gong, Sen Wang, Zi Huang, and Yadan Luo. Is less more? exploring token condensation as training-free test-time adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 144–154, 2025.

Haoyan Wu, Yahao Liu, Yinjie Lei, Lixin Duan, and Wen Li. Dynamic logits adjustment and exploration for test-time adaptation in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3143–3153, 2026a.

Xiangyu Wu, Dongming Jiang, Feng Yu, Yueying Tian, Jiaqi Tang, Qing-Guo Chen, Yang Yang, and Jianfeng Lu. Adaptive debiasing tsallis entropy for test-time adaptation. In *International Conference on Learning Representations*, 2026b.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. URL `https://faculty.cc.gatech.edu/~hays/papers/sun.pdf`.

Jingjing Xie, Yuxin Zhang, Jun Peng, Zhaohong Huang, and Liujuan Cao. Textrefiner: Internal visual feature as efficient refiner for vision-language models prompt tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. ULIP: Learning a unified representation of language, images, and point clouds for 3D understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6757–6767, 2023.

Maxime Zanella and Ismail Ben Ayed. On the test-time zero-shot generalization of vision-language models: Do we really need prompt learning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23783–23793, 2024.

Maxime Zanella, Clément Fuchs, Christophe De Vleeschouwer, and Ismail Ben Ayed. Realistic test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25103–25112, 2025.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. LiT: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18123–18133, 2022.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

Ce Zhang, Simon Stepputtis, Katia Sycara, and Yaqi Xie. Dual prototype evolving for test-time generalization of vision-language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 32111–32136, 2024a.

Jingyi Zhang, Jiaxing Huang, Xiaoqin Zhang, Ling Shao, and Shijian Lu. Historical test-time prompt tuning for vision foundation models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 12872–12896, 2024b.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of CLIP for few-shot classification. In *Computer Vision–ECCV 2022*, pp. 493–510, 2022.

Taolin Zhang, Jinpeng Wang, Hang Guo, Tao Dai, Bin Chen, and Shu-Tao Xia. Boostadapter: Improving vision-language test-time adaptation via regional bootstrapping. In *Advances in Neural Information Processing Systems*, volume 37, 2024c.

Youjia Zhang, Youngeun Kim, Young-Geun Choi, Hongyeob Kim, Huiling Liu, and Sungeun Hong. Backpropagation-free test-time adaptation via probabilistic gaussian alignment. In *Advances in Neural Information Processing Systems*, volume 38, pp. 76280–76310, 2025.

Yuchao Zhang, Hao Wang, Fan Zhang, Qirui Mi, Mengyue Yang, Yisen Wang, Jun Wang, Haoxuan Li, and Zhouchen Lin. Learning to memorize with attributive and associative memory for online test-time adaptation of vision-language models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16816–16825, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130:233–250, 2022b.

Lihua Zhou, Mao Ye, Shuaifeng Li, Nianxin Li, Xiatian Zhu, Lei Deng, Hongbin Liu, and Zhen Lei. Bayesian test-time adaptation for vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29999–30009, 2025.

A Theoretical Analysis

A.1 Hard-Allocation Risk under an Offset-Aligned Correction

Let $(X, Y) \sim \mathcal{D}$, with input marginal $\mathcal{D}_X$. Assume that, for $\mathcal{D}_X$-almost every x, $p_c^\star(x) = \Pr_{\mathcal{D}}(Y = c \mid X = x) > 0$ for every class c, and

$$z_c(x) = \log p_c^\star(x) + \beta_c + \zeta(x), \tag{14}$$

where $\beta \in \mathbb{R}^C$ is class-dependent and $\zeta(x)$ is independent of the class index. Consider a fixed correction $\gamma \in \mathbb{R}^C$ satisfying

$$0 \le \gamma_c - \gamma_j \le \beta_c - \beta_j, \qquad \text{whenever } \beta_c \ge \beta_j. \tag{15}$$

Proposition 1. Let $\hat{y}(x) = \arg\max_c z_c(x)$ and $\hat{y}_\gamma(x) = \arg\max_c [z_c(x) - \gamma_c]$, using the same fixed class ordering to break ties. Under Eqs. (14)–(15), for D_X-almost every x,

$$p^\star_{\hat{y}_\gamma(x)}(x) \ge p^\star_{\hat{y}(x)}(x). \tag{16}$$

Consequently,

$$\epsilon_{\text{alloc}}(\hat{y}_\gamma) \le \epsilon_{\text{alloc}}(\hat{y}). \tag{17}$$

Proof. Fix x and abbreviate $a = \hat{y}(x)$ and $b = \hat{y}_\gamma(x)$. There is nothing to prove when $a = b$. Otherwise, optimality before and after correction gives

$$0 \le z_a(x) - z_b(x) \le \gamma_a - \gamma_b. \tag{18}$$

If $\gamma_a = \gamma_b$, these inequalities imply $z_a = z_b$, and the corrected scores of a and b are also equal. The unchanged priority ordering would then preclude a change from a to b. Consequently, $\gamma_a > \gamma_b$. If $\beta_a \le \beta_b$, applying Eq. 15 to (b, a) would instead give $\gamma_b \ge \gamma_a$, a contradiction. Thus $\beta_a > \beta_b$, and

$$0 \le z_a - z_b \le \gamma_a - \gamma_b \le \beta_a - \beta_b. \tag{19}$$

The class-independent term $\zeta(x)$ cancels, so

$$\log \frac{p_b^\star(x)}{p_a^\star(x)} = z_b(x) - z_a(x) + \beta_a - \beta_b \ge 0. \tag{20}$$

It follows that $p_b^\star(x) \ge p_a^\star(x)$. Finally,

$$\epsilon_{\text{alloc}}(\hat{y}_\gamma) = \mathbb{E}_X[1 - p^\star_{\hat{y}_\gamma(X)}(X)] \le \mathbb{E}_X[1 - p^\star_{\hat{y}(X)}(X)] = \epsilon_{\text{alloc}}(\hat{y}). \tag{21}$$

This proves Proposition A.1 and the hard-allocation risk inequality in Eq. 7.

Scope. This proposition provides a sufficient condition for non-increasing hard-allocation risk under the idealized posterior model. The implemented history-based correction is not assumed to satisfy Eq. 15 exactly; rather, the result motivates offset-aligned regulation of class allocation.

A.2 Classifier Construction and Common-Energy Analysis

From stream statistics to class-state statistics. Under the additive update $P_t = P_{t-1} + A_t f_t$ with $P_0 = e$ and $\mathbf{1}^\top A_t = 1$, define the mean class state $\bar{v}_t = C^{-1} \sum_c P_{t,c,:}^\top$ and the mean text classifier $\bar{e} = C^{-1} \sum_c e_c$. Summing the state update across classes yields

$$\bar{v}_t = \bar{e} + \frac{1}{C} \sum_{i=1}^{t} f_i = \bar{e} + \frac{t}{C} \mu_t. \tag{22}$$

Thus, the stream-wide mean contributes a shared component to the average accumulated class state under both oracle and predicted simplex-valued allocations.

Optimal direction under the activity-weighted criterion. Let $G \in (0,\infty)^D$, $H = \mathrm{Diag}(G)$, and $s \neq 0$. For any nonzero w,

$$s^\top w = (H^{-1/2}s)^\top (H^{1/2}w) \leq \sqrt{s^\top H^{-1}s}\sqrt{w^\top H w}. \tag{23}$$

Therefore,

$$\frac{s^\top w}{\sqrt{w^\top H w}} \leq \sqrt{s^\top H^{-1}s}, \tag{24}$$

with equality when w is a positive multiple of $H^{-1}s$. Imposing $\|w\|_2 = 1$ gives

$$w^\star = \frac{H^{-1}s}{\|H^{-1}s\|_2} = \frac{s \oslash G}{\|s \oslash G\|_2}. \tag{25}$$

This establishes Eq. 10 for the proposed activity-weighted criterion.

Effect on the pre-normalization common-energy fraction. Let $V \in \mathbb{R}^{C\times D}$ be a nonzero class-state matrix, with rows $v_c^\top$, and define $\bar{v} = C^{-1}\sum_c v_c$. For each coordinate, let

$$U_d = \bar{v}_d^2, \qquad B_d = \frac{1}{C}\sum_c (v_{c,d} - \bar{v}_d)^2, \qquad T_d = U_d + B_d = \frac{1}{C}\sum_c v_{c,d}^2. \tag{26}$$

Write $\rho_{\mathrm{com}} = \sum_d U_d / \sum_d T_d$. On coordinates with $T_d > 0$, define

$$\nu_d = \frac{T_d}{\sum_e T_e}, \qquad r_d = \frac{U_d}{T_d}, \qquad h_d = G_d^{-2}. \tag{27}$$

Coordinates with $T_d = 0$ contribute neither common nor total energy and can be omitted. Under $V' = V\,\mathrm{Diag}(G)^{-1}$, the coordinate energies become $U'_d = h_d U_d$, $B'_d = h_d B_d$, and $T'_d = h_d T_d$. It follows that

$$\rho'_{\mathrm{com}} - \rho_{\mathrm{com}} = \frac{\mathbb{E}_\nu[hr]}{\mathbb{E}_\nu[h]} - \mathbb{E}_\nu[r] = \frac{\mathrm{Cov}_\nu(h,r)}{\mathbb{E}_\nu[h]}. \tag{28}$$

Equation 28 shows that ACC decreases the pre-normalization common-energy fraction exactly when $\mathrm{Cov}_\nu(h,r) < 0$. Since $h_d = G_d^{-2}$ decreases with coordinate activity, this condition is satisfied when highly active coordinates tend to carry above-average common-energy fractions.

Relationship to normalized classifier geometry. For unit-normalized classifier directions,

$$\rho_{\mathrm{com}}(W) = \frac{1 + (C-1)\bar{\kappa}}{C}, \tag{29}$$

where $\bar{\kappa}$ is the mean pairwise cosine similarity. Thus, after normalization, common energy and mean pairwise cosine describe the same geometry. Because ACC acts before row normalization, this covariance analysis characterizes its pre-normalization effect, while the resulting angular separation is evaluated empirically in Fig. 3(b) and Appendix C.2.

B ONLINE INFERENCE PROCEDURE

Algorithm 1 Online inference of UTA

Input: visual encoder Φ_v, text classifier $\boldsymbol{e}$, test stream $\{x_t\}$, allocation size k.

1: **Initialize**: historical response state $\boldsymbol{M}_0 = \boldsymbol{0}$, accumulated class states $\boldsymbol{P}_0 = \boldsymbol{e}$, and coordinate-activity vector $\boldsymbol{G}_0 = \boldsymbol{1}$.
2: **for** each test sample $\boldsymbol{x}_t$ **do**
3: Obtain $\boldsymbol{f}_t$, zero-shot textual logits $\boldsymbol{z}_t^{\mathrm{text}}$ and probabilities $\boldsymbol{p}_t$ by Φ_v and $\boldsymbol{e}$. ▷ Eq. (1)
4: Compute $\boldsymbol{\delta}_t$ from $\boldsymbol{M}_{t-1}$ and obtain $\boldsymbol{q}_t$. ▷ Eq. (3)∼(5)
5: Set $\boldsymbol{A}_t = \mathcal{S}_k(\boldsymbol{q}_t)$, and update $\boldsymbol{P}_t$ and $\boldsymbol{G}_t$. ▷ Eq. (11)
6: Construct $\boldsymbol{W}_t$, compute $\boldsymbol{z}_t^{\mathrm{UTA}}$ and output $\hat{y}_t$. ▷ Eq. (12)∼(13)
7: Update $\boldsymbol{M}_t = \boldsymbol{M}_{t-1} + \boldsymbol{q}_t$.
8: **end for**

C SUPPLEMENTARY EXPERIMENTS

C.1 CLASS-WISE RESPONSE MISMATCH.

Figure 4 extends the DTD example in Figure 1(a) to all ten cross-dataset benchmarks. Across datasets, the mean zero-shot response is generally not aligned with the empirical label frequency, indicating persistent class-wise response preferences rather than a dataset-specific phenomenon. Notably, this mismatch remains visible even on datasets with nearly uniform empirical class frequencies, such as DTD, Food101, and SUN397, suggesting that it cannot be attributed solely to class imbalance. The corresponding TV distances range from 5.8% to 20.4%, further illustrating that the magnitude of the mismatch varies across datasets.

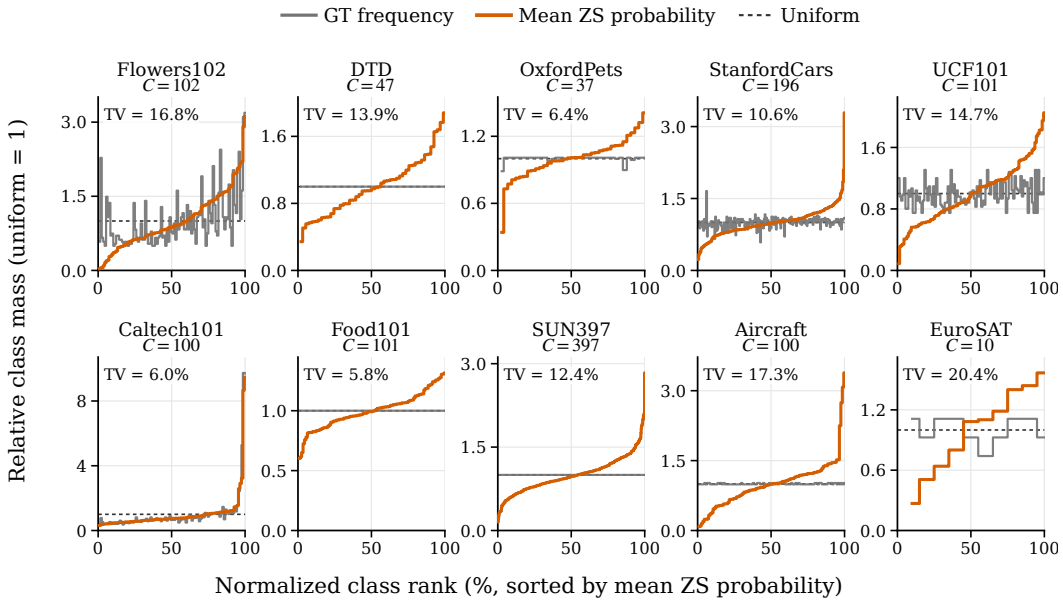


Figure 4: Class-wise mismatch between zero-shot responses and empirical label frequencies on the ten cross-dataset benchmarks. Within each dataset, classes are sorted by their mean raw zero-shot probability, and the ground-truth frequencies are reordered using the same class indices. Both curves are normalized by the uniform class mass $1/C$, so the dotted horizontal line corresponds to a uniform distribution. The reported TV value measures the total-variation distance between the mean zero-shot response and the empirical label distribution.

C.2 DATASET-WISE MECHANISM DIAGNOSTICS

We report the dataset-wise statistics underlying the mechanism analysis in Figure 3. The diagnostics are separated into class allocation and classifier construction to match the two failure modes studied in the paper. For class allocation, data imbalance $\mathrm{TV}(\pi, u) = \frac{1}{2}\|\pi - u\|_1$. measures the total-variation distance between the empirical label distribution and the uniform distribution, while prediction bias $\mathrm{TV}(\bar{p}, \pi) = \frac{1}{2}\|\bar{p} - \pi\|_1$ measures the distance between the mean zero-shot response and the empirical label distribution (denoted as Mismatch). The history–future correlation is computed within each dataset across classes using first-half raw zero-shot responses and second-half prediction excess. The allocation-accuracy comparison is a zero-shot-only diagnostic using the same corrected-history update as HAD, while removing the visual-classifier branch to isolate the effect of historical regulation. For classifier construction, all paired comparisons use identical class assignments. Predicted-allocation rows evaluate the visual classifier without text-logit fusion. For oracle visual accuracy, the classifier used to predict sample t is constructed only from ground-truth assignments of preceding samples $1, \ldots, t-1$; the current label y_t is never used before prediction. This allows the contribution of ACC to classifier construction to be examined independently of HAD. Figure 4 shows the class-wise

Table 7: Dataset-wise mechanism diagnostics on the ten cross-dataset benchmarks. TV denotes total-variation distance. History–future correlation is the class-wise Spearman correlation between first-half mean zero-shot responses and second-half prediction excess. Allocation accuracy is measured in a zero-shot-only, past-only HAD diagnostic. Classifier-construction comparisons use matched assignments, and visual accuracy excludes final text-logit fusion.

Metric	Flowers	DTD	Pets	Cars	UCF101	Caltech	Food101	SUN397	Aircraft	EuroSAT	**Avg.**
I. Class allocation											
Data imbalance (TV, %)	20.11	0.00	0.69	3.71	6.30	27.85	0.00	0.00	0.66	5.56	6.49
Mismatch (TV, %)	16.78	13.95	6.41	10.58	14.73	6.01	5.79	12.38	17.34	20.42	12.44
History-future corr. (ρ)	0.53	0.94	0.84	0.75	0.86	0.05	0.92	0.90	0.89	0.95	0.76
ZS assignment acc. (%)	70.73	44.09	89.07	66.21	67.54	**93.91**	85.90	66.31	24.72	48.21	65.67
+ HAD	**71.46**	**46.04**	**90.87**	**67.83**	**72.48**	92.58	**86.38**	**68.80**	**26.82**	**60.69**	**68.39**
II. Classifier construction											
Pred. visual acc. (%)	73.57	46.28	89.62	68.26	73.96	93.47	84.16	68.75	26.31	62.81	68.72
+ ACC	**74.79**	**48.35**	**91.01**	**69.89**	**75.10**	**94.12**	**85.32**	**70.23**	**26.73**	**65.49**	**70.10**
Final nearest angle (°)	18.4	18.7	14.4	14.8	22.1	27.6	20.1	21.4	13.1	11.9	18.3
+ ACC	**37.9**	**44.4**	**31.0**	**26.3**	**41.0**	**51.4**	**40.9**	**39.4**	**24.5**	**30.5**	**36.8**
Oracle visual acc. (%)	83.84	58.22	85.17	68.60	80.73	86.94	82.88	67.57	**37.56**	77.40	72.89
+ ACC	**86.16**	**62.53**	**87.87**	**72.67**	**83.74**	**90.47**	**84.46**	**71.37**	36.96	**82.15**	**75.84**

Table 8: Full component ablation on the ten cross-dataset benchmarks. All variants use CLIP ViT-B/16, the same CLIP prompt ensemble, and paired seed-1 test streams. The first row is zero-shot CLIP; "Visual" denotes the accumulated P_t-based visual classifier with standard row normalization. Checkmarks indicate enabled HAD or ACC components. Values are top-1 accuracy (%), and Avg. is the unweighted mean over datasets.

Visual $\boldsymbol{W}$	HAD	ACC	Flowers	DTD	Pets	Cars	UCF	Cal.	Food	SUN	Air.	Euro.	Avg.
			70.73	44.09	89.07	66.21	67.54	93.91	85.90	66.31	24.72	48.21	65.67
✓			72.43	46.69	90.81	68.19	69.71	94.16	86.35	68.14	24.87	49.72	67.11
✓	✓		73.97	48.11	91.44	69.84	74.62	93.96	86.70	70.12	**27.36**	63.43	69.96
✓		✓	72.88	46.99	91.25	68.46	70.13	**94.40**	86.34	68.73	24.78	50.98	67.49
✓	✓	✓	**74.42**	**49.23**	**92.07**	**70.51**	**75.68**	94.24	**86.79**	**70.93**	27.03	**65.64**	**70.66**

response mismatch across all ten datasets; the corresponding temporal and classifier-construction diagnostics are reported in Table 7.

C.3 Prompt Robustness

To further examine the robustness of UTA to different textual representations, we evaluate its performance under three prompt constructions with increasing levels of prompt engineering: a single vanilla template, the original CLIP template ensemble, and GPT/CuPL class descriptions. As shown in Table 9, UTA consistently achieves the strongest average performance across all three settings, reaching 69.44%, 70.66%, and 71.11%, respectively. A notable observation is that the performance of several adaptation methods varies considerably with the prompt construction. In contrast, UTA consistently improves over the corresponding zero-shot CLIP baseline and the paired statistics-based methods across all three settings. In particular, UTA remains effective even with the single vanilla template, while further benefiting from richer textual representations. These results indicate that the proposed allocation debiasing and classifier construction mechanisms do not rely on a specific prompt ensemble or externally generated class descriptions.

C.4 Component Ablations

We provide the full per-dataset results corresponding to the main-text component ablation. All variants are evaluated on identical ViT-B/16 test streams and use the same CLIP prompt ensemble. The first row is the pure zero-shot CLIP baseline. The second row adds the accumulated visual

Table 9: Prompt robustness on the ten cross-dataset benchmarks using CLIP ViT-B/16. We compare a single vanilla template, the original CLIP template ensemble, and GPT/CuPL class descriptions. Within each prompt block, paired statistics-based methods share the same text classifier and seed-1 test streams. Values are top-1 accuracy (%), Avg. is the unweighted mean, and bold marks the best paired statistics-based method in each block.

Method	Flowers	DTD	Pets	Cars	UCF	Cal.	Food	SUN	Air.	Euro.	Avg.
Vanilla (`a photo of a [class]`) (Radford et al., 2021)											
CLIP	67.15	44.39	88.01	65.29	65.00	92.94	85.41	62.59	23.79	41.44	63.60
TPT	68.98	47.75	87.79	68.50	68.04	94.16	84.67	65.50	24.78	42.44	65.26
DiffTPT	70.10	47.00	88.22	67.01	62.67	92.49	87.23	65.74	25.60	43.13	64.92
TDA	69.87	46.16	88.74	66.68	68.62	93.43	85.93	65.99	24.36	53.02	66.28
BCA†	67.11	44.68	87.60	65.83	65.32	93.02	85.16	63.13	23.88	36.99	63.27
SCA†	**73.57**	48.17	89.13	67.43	69.65	93.39	85.61	67.41	24.99	54.63	67.40
PTA†	72.47	47.34	89.42	68.46	69.89	**94.08**	85.93	67.19	24.93	56.35	67.61
UTA	73.00	**48.58**	**90.57**	**70.36**	**72.56**	93.10	**86.77**	**69.99**	**26.43**	**62.99**	**69.44**
CLIP Templates (Radford et al., 2021)											
CLIP†	70.73	44.09	89.07	66.21	67.54	93.91	85.90	66.31	24.72	48.21	65.67
TDA†	71.21	44.86	89.67	67.43	70.55	94.08	86.14	67.84	25.32	61.75	67.89
BCA†	70.44	44.21	88.85	66.32	67.88	93.96	85.92	66.50	25.20	49.57	65.89
SCA†	74.26	48.40	90.11	67.72	71.11	93.75	86.06	68.96	25.14	49.58	67.51
PTA†	**74.91**	47.22	91.33	68.65	73.22	**94.65**	86.34	69.12	25.77	61.36	69.26
UTA	74.42	**49.23**	**92.07**	**70.51**	**75.68**	94.24	**86.79**	**70.93**	**27.03**	**65.64**	**70.66**
GPT/CuPL Class Descriptions (Zhang et al., 2024a; Zhou et al., 2025)											
CLIP	72.76	53.25	90.19	65.95	66.69	94.24	85.85	67.93	27.21	53.60	67.77
TDA	74.02	54.61	90.95	66.80	69.79	94.48	86.12	68.92	28.59	54.88	68.92
DPE	75.07	54.20	91.14	67.31	70.44	94.81	86.17	70.07	28.95	55.79	69.40
BCA†	72.96	53.43	90.35	66.75	67.70	94.44	85.93	68.35	28.47	56.65	68.50
SCA†	76.65	**56.32**	91.63	67.47	69.92	94.20	86.03	70.18	28.50	57.17	69.81
PTA†	77.18	55.61	91.47	67.94	70.82	**94.93**	86.40	70.56	28.65	54.67	69.82
UTA	**77.22**	55.20	**92.48**	**70.00**	**72.80**	94.44	**86.76**	**72.01**	**30.24**	**59.93**	**71.11**

classifier constructed from P_t using standard row normalization, while disabling both HAD and ACC. The remaining rows enable HAD, ACC, or both on top of this common visual-classifier baseline. Table 8 shows that HAD provides the larger individual gain, increasing the macro average from 67.11% to 69.96%. ACC alone gives a smaller improvement to 67.49%, while adding ACC on top of HAD further raises the average to 70.66% and improves nine of the ten datasets.

C.5 Plug-in Transfer to Existing Methods

We further examine whether the two UTA components can be integrated into existing TTA pipelines without replacing their native adaptation mechanisms. All paired comparisons use matched seed-1 streams and the same CLIP prompt ensemble. For the statistics-based methods BCA, SCA, and PTA, we integrate both HAD and ACC. HAD corrects the logits entering each method's allocation or prediction pathway, while ACC conditions its visual-classifier readout without modifying the native state updates. For TDA, ADAPT, DPE, and TCA, only HAD is transferred because their internal representations do not expose the same accumulated class-state structure required by ACC.

Table 10 shows that all paired baselines benefit in macro-average accuracy after the corresponding UTA components are introduced. The gains across heterogeneous adaptation mechanisms support the use of HAD and ACC as lightweight plug-in components rather than as operations specific to the proposed base classifier.

C.6 Backbone Transfer to CLIP ResNet-50

We evaluate whether UTA transfers beyond the ViT-B/16 backbone used in the main experiments. Without modifying the UTA update rules, we replace the CLIP visual encoder with ResNet-50 and

Table 10: Transfer of UTA components to existing TTA methods on the ten cross-dataset benchmarks. BCA, SCA, and PTA receive both HAD and ACC, whereas TDA, ADAPT, DPE, and TCA receive HAD only. All paired rows use CLIP ViT-B/16, matched seed-1 streams, and the same CLIP prompt ensemble. Values are top-1 accuracy (%); Avg. is the unweighted mean over datasets. The insertion configuration for each baseline is selected by its ten-dataset mean and then fixed across datasets.

Method	Flowers	DTD	Pets	Cars	UCF101	Caltech	Food101	SUN397	Aircraft	EuroSAT	Average
BCA	70.44	44.21	88.85	66.32	67.88	93.96	85.92	66.50	25.20	49.57	65.89
+UTA	71.09	46.22	90.68	67.33	72.61	92.62	85.80	68.34	27.81	60.15	**68.26**
SCA	74.26	48.40	90.11	67.72	71.11	93.75	86.06	68.96	25.14	49.58	67.51
+UTA	73.61	48.76	91.85	70.28	75.10	92.41	86.50	70.45	26.31	63.57	**69.88**
PTA	74.91	47.22	91.33	68.65	73.22	94.65	86.34	69.12	25.77	61.36	69.26
+UTA	75.31	48.58	91.93	70.74	75.84	94.36	86.81	71.12	27.54	61.78	**70.40**
TDA	71.21	44.86	89.67	67.43	70.55	94.08	86.14	67.84	25.32	61.75	67.89
+UTA	71.74	47.46	89.86	68.34	73.65	93.55	86.24	69.03	26.88	60.63	**68.74**
ADAPT	73.04	47.04	91.36	69.01	70.63	93.23	85.56	69.08	24.93	64.41	68.83
+UTA	73.12	46.39	91.96	70.35	74.09	91.85	86.05	70.29	27.03	62.67	**69.38**
DPE	72.51	45.57	90.19	68.05	72.91	94.85	86.47	68.35	26.55	57.32	68.28
+UTA	73.20	46.75	91.20	68.66	73.54	94.77	86.66	69.38	26.79	63.17	**69.41**
TCA	73.08	44.03	89.07	65.54	69.28	93.87	84.61	67.03	24.51	66.63	67.77
+UTA	73.65	46.87	90.79	66.67	73.86	94.12	85.72	69.00	27.30	67.53	**69.55**

Table 11: Cross-dataset generalization with CLIP ResNet-50. Values are top-1 accuracy (%) on the ten benchmarks and their unweighted mean.

Method	Flowers	DTD	Pets	Cars	UCF	Cal.	Food	SUN	Air.	Euro.	Avg.
CLIP (Radford et al., 2021)	66.10	42.20	85.83	56.55	61.41	87.91	76.34	61.11	17.22	37.42	59.21
Backpropagation-Based Test-Time Adaptation											
TPT (Shu et al., 2022)	62.69	40.84	84.49	58.46	60.82	87.02	74.88	61.46	17.58	28.33	57.66
DiffTPT (Feng et al., 2023)	63.53	40.72	83.40	60.71	62.67	86.89	79.21	62.72	17.60	41.04	59.85
DPE (Zhang et al., 2024a)	67.60	50.18	85.97	59.26	61.98	90.83	77.83	64.23	19.80	41.67	61.93
Backpropagation-Free Test-Time Adaptation											
TDA (Karmanov et al., 2024)	**68.70**	43.62	86.15	57.39	63.81	**89.70**	77.74	62.37	17.40	41.23	60.81
ADAPT (Zhang et al., 2025)	67.97	45.57	87.76	59.68	63.49	88.24	76.35	63.77	17.22	47.44	61.75
BCA (Zhou et al., 2025)	66.26	42.38	85.45	58.13	61.80	87.75	77.39	61.36	17.28	29.04	58.68
SCA (Guan et al., 2025)	67.72	**46.69**	86.43	59.23	66.03	88.40	76.72	63.17	17.01	48.51	61.99
PTA (Huang et al., 2026)	68.49	44.98	86.67	59.35	66.43	88.64	76.83	62.86	17.44	**51.63**	62.33
UTA	68.25	46.51	**88.74**	**61.04**	**68.38**	88.07	**78.19**	**65.47**	**18.93**	49.01	**63.26**

follow the prompt protocol used by PTA. As shown in Table 11, UTA achieves the highest average among the compared backpropagation-free methods and improves over PTA on seven of the ten cross-dataset benchmarks.

C.7 Robustness to test-stream ordering.

Since online adaptation may be affected by the arrival order of test samples, we further examine the sensitivity of our method to random stream permutations. Specifically, three test streams are independently generated with different random seeds while keeping all other settings unchanged. As shown in Table 12, the performance remains highly consistent across different sequences, yielding average accuracies of 70.66%, 70.63%, and 70.58%, respectively. The resulting mean accuracy is 70.62%, with a standard deviation of only 0.04 across the ten datasets. Even at the dataset level, the variation remains small, indicating that the proposed adaptation is robust to ordinary changes in test-sample ordering rather than relying on a favorable stream arrangement.

C.8 RESULTS ON NON-IID TEST STREAMS.

We further evaluate robustness under strongly class-correlated test streams following the protocol of PTA. Test sequences are generated using a Dirichlet distribution, where the concentration parameter γ controls the degree of non-IIDness, with smaller values inducing stronger class concentration. We consider $\gamma = 0.1$, 0.01, and 0.001, together with the more challenging Separate setting in which classes arrive sequentially. As reported in Table 13, our method consistently outperforms PTA across all four configurations, improving the average accuracy from 67.29% to 68.95%, 66.79% to 67.90%, and 66.65% to 67.63% as γ decreases from 0.1 to 0.001. The advantage is also retained under the extreme Separate setting, achieving 67.22% compared with 66.33% for PTA. These results indicate that the proposed adaptation remains effective even when historical observations are strongly biased toward recently occurring classes, supporting its robustness to temporally correlated test distributions.

Table 12: Robustness to test-stream ordering. Accuracy (%) over three independently shuffled test sequences on the 10 cross-domain datasets, generated with random seeds 1, 2, and 3, respectively. The mean and standard deviation are reported across the three sequences. The consistently small variance indicates that the proposed method is insensitive to random test-stream ordering.

Sequence	Flowers	DTD	Pets	Cars	UCF	Cal.	Food	SUN	Air.	Euro.	Avg.
Seq. 1	74.42	49.23	92.07	70.51	75.68	94.24	86.79	70.93	27.03	65.64	70.66
Seq. 2	74.67	47.87	92.12	69.95	76.29	93.83	86.99	70.90	27.54	66.09	70.63
Seq. 3	74.83	48.82	91.82	70.53	75.34	93.79	86.88	70.73	26.82	66.23	70.58
Mean	74.64	48.64	92.01	70.33	75.77	93.96	86.89	70.85	27.13	65.99	70.62
Std.	0.20	0.70	0.16	0.33	0.48	0.25	0.10	0.11	0.37	0.31	0.04

Table 13: Performance under non-IID test streams. Accuracy (%) on ImageNet and 10 cross-domain datasets under different degrees of class-correlated test streams. Following PTA, non-IID streams are constructed using a Dirichlet distribution with $\gamma \in \{0.1, 0.01, 0.001\}$, where smaller γ indicates stronger non-IIDness. Separate denotes the extreme setting in which samples from different classes arrive sequentially. For a fair comparison, we adopt the prompt templates described in the original CLIP (Radford et al., 2021), where † denotes results reproduced using the officially released code. The best result in each setting is shown in **bold.**

Scenario	Method	ImageNet	Flowers	DTD	Pets	Cars	UCF	Cal.	Food	SUN	Air.	Euro.	Avg.
–	CLIP†	66.75	70.73	43.44	89.10	65.58	67.49	93.31	85.91	62.55	24.78	48.38	65.27
Low ($\gamma = 0.1$)	PTA†	68.07	72.67	44.21	90.46	67.58	71.08	93.42	86.18	66.03	22.17	58.34	67.29
	UTA	**70.22**	**74.11**	**47.40**	**91.40**	**69.05**	**72.06**	**93.51**	**86.29**	**68.44**	**25.61**	**60.38**	**68.95**
Medium ($\gamma = 0.01$)	PTA†	68.22	72.77	44.32	90.07	67.39	70.38	**93.37**	**85.56**	65.97	22.20	54.49	66.79
	UTA	**69.50**	**74.14**	**46.08**	**90.86**	**68.52**	**71.16**	93.19	85.36	**67.31**	**25.22**	**55.60**	**67.90**
High ($\gamma = 0.001$)	PTA†	68.28	73.26	44.59	89.63	66.92	69.77	93.16	**85.11**	65.65	22.61	54.22	66.65
	UTA	**68.76**	**74.40**	**45.93**	**91.00**	**68.21**	**70.65**	**93.18**	84.85	**66.57**	**25.12**	**55.28**	**67.63**
Separate	PTA†	68.23	72.66	45.05	89.37	66.97	70.04	**93.51**	**84.96**	65.49	22.50	**50.89**	66.33
	UTA	**68.60**	**74.37**	**46.22**	**90.61**	**68.24**	**70.82**	93.30	84.74	**66.31**	**25.44**	50.72	**67.22**

D LIMITATIONS

UTA regulates class allocation and classifier construction using compact statistics accumulated from the test stream. HAD specifically mitigates persistent class-wise prediction preferences that are reflected in historical responses, while ACC improves classifier construction from the resulting class states. Nevertheless, fixed-size aggregate statistics may not fully capture other forms of prediction bias that do not manifest as stable class-wise response patterns. The larger improvement of ACC under oracle allocation than under predicted allocation also suggests that residual allocation errors can limit the potential of subsequent classifier conditioning. Since individual contributions cannot be selectively recovered once incorporated into the aggregate class states, exploring richer yet compact statistics for capturing such residual biases is a promising direction for future work.