Hyperspherical Prototype-Driven Feature Adaptation for Test-Time Adaptation under Long-Tailed Pretraining

Anonymous authors
Paper under double-blind review

Abstract

Test-Time Adaptation (TTA) techniques aim to adapt a pretrained model to an unlabeled target domain during inference, and existing methods are typically designed under the assumption of relatively balanced source training data. When the source dataset is long-tailed or highly-imbalanced, the classifier bias inherited from source training can cause conventional TTA to further favor majority classes and degrade minority-class recognition, even when the target distribution is balanced. To address this technical gap, we formulate TTA under long-tailed source pretraining as a *prototype-driven feature adaptation* problem, in which target feature representations should be adapted to better fit the decision boundaries learned from the long-tailed source data. Accordingly, we propose Hyperspherical Prototype-Driven Feature Adaptation TTA (HyPFA-TTA). HyPFA-TTA uses the normalized source classifier weights as fixed class prototypes and estimates the class tendency of each incoming target batch from the distance-based associations between target features and fixed source prototypes on a unit hypersphere. The estimated tendency is then used to correct source-biased predictions before pseudo-label generation. These corrected pseudo-labels provide prototype-based supervision for the cross-entropy and prototype-alignment objectives, through which the trainable parameters of the batch normalization layers are updated during adaptation. Experiments on ImageNet-LT→ImageNet-C and the real-world Whitebait-31→Whitebait-31Ex dataset demonstrate that HyPFA-TTA consistently outperforms existing TTA methods across different corruption types and source class-frequency groups, with improvements extending to both head and non-head classes.

1 Introduction

Test-Time Adaptation (TTA) adapts a pretrained model to unlabeled target samples during inference without accessing the source training data Liang et al. (2025). Existing TTA methods address target distribution shifts through entropy minimization and normalization adaptation Li et al. (2017); Wang et al. (2021); Niu et al. (2022; 2023), prediction refinement and consistency learning Liang et al. (2020); Wang et al. (2022); Yuan et al. (2023), and prototype- or representation-based adaptation Iwasawa & Matsuo (2021); Chen et al. (2022). However, these methods commonly assume a relatively balanced source dataset. When the source training data are long-tailed or highly-imbalanced, the pretrained model inherits class-dependent biases that can directly affect subsequent adaptation. In particular, long-tailed training biases predictions toward source-majority classes Kang et al. (2020) and induces class-dependent variations in feature and classifier weight norms Alshammari et al. (2022); Dang et al. (2023). Consequently, these source-induced biases can affect pseudo-label estimation, driving subsequent adaptation further toward majority classes and degrading minority-class recognition even when the target dataset is balanced. TTA techniques for models pretrained on long-tailed or highly-imbalanced data must therefore address the target distribution shift without allowing the source-induced class bias to dominate the adaptation process.

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

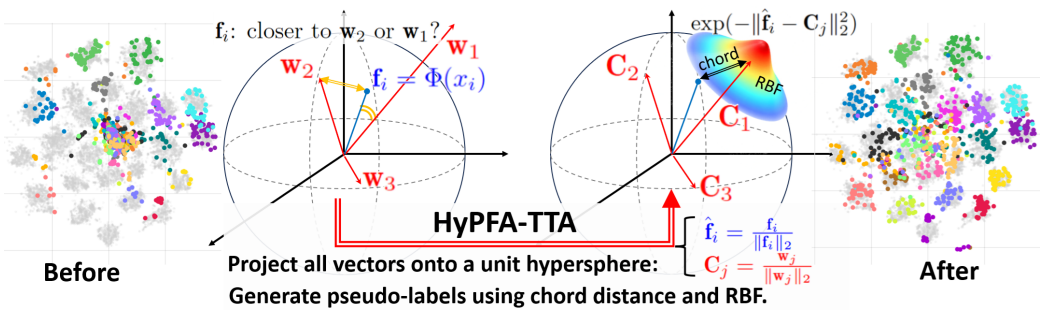


Figure 1: **Overview of HyPFA-TTA for test-time adaptation under long-tailed source pretraining.** A source classification model pretrained on long-tailed data can produce initial target predictions biased toward head classes, leading to biased pseudo-label estimation and subsequently driving adaptation further toward the head classes. We associate this problem with the class-dependent variations in classifier-weight and feature-vector norms induced by long-tailed pretraining. HyPFA-TTA therefore projects the unequal-norm classifier weights $\mathbf{w}_j$, which serve as source class prototypes, and target feature vectors onto a unit hypersphere. On the hypersphere, chord distances and a radial basis function (RBF) are used to measure the association between each target sample feature $\hat{\mathbf{f}}_i$ and the source prototypes $\mathbf{C}_j$, from which more stable pseudo-labels are estimated. Target features are thus adapted to better align with their corresponding source prototypes on the hypersphere. The two t-SNE maps correspond to the experiments reported in the lower-right parts of Table 2, and gray points denote source features. (Refer to Fig. 2 in the Appendix for t-SNE settings.)

Despite these source-induced biases, target samples from the same classes remain clustered before TTA, although their feature distribution deviates from that of the source domain, as shown in the left t-SNE map in Fig. 1. This motivates us to adapt the target features to better fit the decision boundaries learned from the long-tailed source data. Since source training samples are inaccessible during TTA, we use the pretrained classifier weights as fixed source class prototypes, *i.e.*, reference anchors that retain source-side class information. Because long-tailed source training induces class-dependent variations in classifier weight norms Alshammari et al. (2022); Dang et al. (2023), we normalize both source prototypes and target features onto a unit hypersphere Liu et al. (2017); Wang et al. (2018); Deng et al. (2019), where their distance-based associations are estimated. Aggregating these soft prototype associations over an incoming target batch provides an estimate of its class tendency, which is subsequently used to correct source-induced prediction bias before pseudo-label generation.

Based on this formulation, we propose **HyPFA-TTA**, a hyperspherical prototype-driven feature adaptation framework for test-time adaptation under long-tailed pretraining. HyPFA-TTA first estimates the batch-level class tendency and incorporates it into pseudo-label generation to reduce the influence of source-majority bias. The estimated pseudo-labels then provide supervision via a cross-entropy loss and a prototype-alignment objective. By updating the trainable parameters of the batch normalization layers with these objectives, HyPFA-TTA progressively adapts the target feature representations to better fit the decision boundaries associated with the source prototypes, while the source prototypes remain fixed.

Our main contributions are summarized as follows:

- We formulate test-time adaptation under long-tailed source pretraining as a prototype-driven feature adaptation problem, in which fixed source class prototypes serve as reference anchors for adapting target features to better fit the decision boundaries associated with these prototypes.
- We propose HyPFA-TTA, which uses normalized source classifier weights as fixed reference prototypes to estimate pseudo-labels for supervising target feature adaptation. HyPFA-TTA reduces the influence of source-majority bias during pseudo-label estimation and uses the resulting pseudo-labels in the cross-entropy and prototype-alignment objectives to adapt the target feature representations.
- We evaluate HyPFA-TTA on ImageNet-LT→ImageNet-C and the real-world datasets:

Whitebait-31→Whitebait-31Ex. The results show consistent improvements over existing TTA methods across different class groups, with the benefits extending beyond the head classes.

2 Related Works

2.1 Test-Time Adaptation

Existing TTA methods perform adaptation using the predictions or representations of incoming unlabeled target samples. AdaBN Li et al. (2017) adapts Batch Normalization statistics, while Tent Wang et al. (2021) minimizes prediction entropy by optimizing Batch Normalization affine parameters. EATA Niu et al. (2022) and SAR Niu et al. (2023) further improve entropy-based adaptation through sample selection, anti-forgetting regularization, and sharpness-aware optimization. SHOT Liang et al. (2020) combines information maximization with pseudo-label refinement, CoTTA Wang et al. (2022) uses teacher-student predictions for continual adaptation, MEMO Zhang et al. (2022) optimizes prediction consistency across augmented views, and AdaContrast Chen et al. (2022) exploits target representations through contrastive learning. T3A Iwasawa & Matsuo (2021) constructs target-aware prototypes from confident samples without updating the feature extractor, whereas LAME Boudiaf et al. (2022) refines predictions without model parameter updates. Despite these different mechanisms, their adaptation is largely driven by predictions or representations produced by the source-pretrained model.

This dependence becomes important when class distributions are imbalanced. Existing studies have mainly investigated imbalance that arises from the target data during deployment. Label Shift Adapter Park et al. (2023) estimates changes in the target label distribution and adjusts predictions accordingly, while ROID Marsden et al. (2024) incorporates prior correction together with diversity weighting and weight ensembling under diverse test distributions. SAR Niu et al. (2023) further shows that entropy-based adaptation can become unstable under online imbalanced label distributions. These studies demonstrate that the class distribution of incoming target samples can substantially affect the behavior of TTA.

Our setting differs in where the class imbalance originates. We consider a model pretrained on a long-tailed source distribution, such that class-dependent bias is present in the pretrained model. Consequently, even when the target distribution itself is balanced, the predictions used to drive adaptation can remain biased toward source-majority classes and further propagate this bias during TTA. We therefore prevent the bias inherited from long-tailed source pretraining from dominating target adaptation through prototype-driven feature adaptation using fixed source class prototypes as reference anchors.

2.2 Long-tailed Representation Learning

Long-tailed recognition addresses learning from highly-imbalanced class distributions, which can bias feature representations and classifiers toward majority classes Liu et al. (2019). Long-tailed training can also induce class-dependent variations in feature and classifier weight norms Alshammari et al. (2022); Dang et al. (2023). Existing methods address these effects through class re-weighting, classifier re-balancing, and prior-aware logit correction Cui et al. (2019); Kang et al. (2020); Menon et al. (2021), typically during source training when source data or class-frequency information remain available.

In contrast, our setting considers the subsequent adaptation of an already trained long-tailed model, where source samples and class frequencies are unavailable and the model has already inherited class-dependent effects from source pretraining. The objective is therefore to adapt unlabeled target features without further reinforcing these inherited biases. This distinction motivates our use of fixed source classifier weights as class prototypes to guide target feature adaptation.

2.3 Prototype and Hyperspherical Representation Learning

Prototype-based learning represents classes by points in the feature space and performs prediction according to feature-to-prototype similarity. In TTA, T3A Iwasawa & Matsuo (2021) constructs target-aware prototypes from confident target samples, while AdaContrast Chen et al. (2022) exploits target representation structure through contrastive learning. In contrast, HyPFA-TTA retains source classifier weights as fixed class prototypes that serve as reference anchors for target feature adaptation. Hyperspherical representation learning provides a suitable space for measuring feature-to-prototype relationships by separating vector magnitude from directional information. SphereFace Liu et al. (2017), CosFace Wang et al. (2018), and ArcFace Deng et al. (2019) normalize features and classifier weights and introduce angular margins to improve class separation. In our setting, hyperspherical normalization instead removes class-dependent vector norm variations from feature-to-prototype matching under long-tailed source pretraining Alshammari et al. (2022); Dang et al. (2023).

Unlike angular-margin methods, HyPFA-TTA does not modify source training or introduce an angular-margin objective. Instead, normalized source classifier weights remain fixed during TTA and serve as reference prototypes for target feature adaptation.

3 HyPFA-TTA: Hyperspherical Prototype-Driven Feature Adaptation for Test-Time Adaptation

We formulate TTA under long-tailed source pretraining as a prototype-driven feature adaptation problem, where target features are adapted to better fit the classification boundaries associated with the source prototypes. HyPFA-TTA uses the normalized source classifier weights as fixed class prototypes, *i.e.*, reference anchors, and performs adaptation in three steps. First, target features and source prototypes are projected onto a unit hypersphere to remove class-dependent vector norm variations caused by long-tailed pretraining. Second, the relative distances between each within-batch target feature and the prototype anchors are then used to estimate the class tendency of each incoming target batch. Third, the estimated tendency is incorporated into pseudo-label generation to reduce the influence of source-majority bias on target predictions. The estimated pseudo-labels are subsequently used in the cross-entropy and prototype-alignment objectives (*i.e.*, $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{align}}$, respectively) to adapt the normalization layers through their trainable affine parameters. These updates progressively affect the feature representations of subsequent target samples, while the source prototypes remain fixed throughout adaptation. Because the pseudo-labels determine which fixed source prototype each target feature should be aligned with, we refer to this indirect adaptation mechanism as **prototype-driven feature adaptation**. Note that the pseudo-code and the data-flow overview of HyPFA-TTA are provided in Appendix A.1.

3.1 Source Prototypes as Reference Anchors for Feature Adaptation

Let the source-pretrained model consist of a feature extractor Φ and a linear classifier with weights $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]^\top$, where $\mathbf{w}_k$ denotes the classifier weight of class k. Since source training samples are unavailable during TTA, we use these classifier weights as source class prototypes that provide fixed reference anchors for feature adaptation.

Long-tailed source pretraining induces class-dependent variations in classifier weight norms Alshammari et al. (2022); Dang et al. (2023). Directly using the original classifier weights as prototypes would therefore retain these class-dependent vector norm variations. We further observe such class-dependent vector norm variations in the source model trained on the real-world Whitebait-31 dataset, as detailed in Appendix Sec. A.4. We remove this effect by normalizing each classifier weight and define the resulting source prototype as $\mathbf{C}_k = \mathbf{w}_k / \|\mathbf{w}_k\|_2$, where $\mathbf{C}_k$ denotes the normalized source prototype of class k, and $\mathbf{C} = [\mathbf{C}_1, \ldots, \mathbf{C}_K]^\top$ denotes the normalized prototype matrix. All source prototypes remain fixed throughout test-time adaptation. Consequently, HyPFA-TTA preserves the class directions learned by the source classifier while preventing class-dependent vector norm variations from affecting the subsequent prototype-based estimation and adaptation.

3.2 Prototype-Driven Class-Tendency Estimation

The main purpose of this stage is to estimate the influence of each source class prototype $\mathbf{C}_k$, which serves as an anchor, at the feature-space position of a target sample $\mathbf{x}_i$. This design follows the intuition commonly used in label propagation and spatial warping, where the influence of an anchor on a sample decreases as the distance between them increases. To model such distance-dependent influence, we employ a Gaussian radial basis function (RBF), whose response decays exponentially with the feature-to-prototype distance. Based on these RBF responses, we estimate the sample-wise prototype association score $a_{i,k}$, which subsequently serves as the basic quantity for estimating the batch-level class tendency π_k^{batch}.

Specifically, we define the sample-wise prototype association score of the target sample $\mathbf{x}_i$ based on the RBF responses as $a_{i,k} = \exp(-\|\hat{\mathbf{f}}_i - \mathbf{C}_k\|_2^2) / \sum_{j=1}^{K} \exp(-\|\hat{\mathbf{f}}_i - \mathbf{C}_j\|_2^2)$, where $\hat{\mathbf{f}}_i = \mathbf{f}_i / \|\mathbf{f}_i\|_2$ with $\mathbf{f}_i = \Phi(\mathbf{x}_i)$ denoting the feature of $\mathbf{x}_i$. Since both vectors $\hat{\mathbf{f}}_i$ and $\mathbf{C}_j$ lie on the unit hypersphere, $\|\hat{\mathbf{f}}_i - \mathbf{C}_j\|_2^2$ corresponds to the squared chord distance between them and monotonically reflects their angular separation, thereby measuring their relative proximity on the hypersphere. Here, $\exp(-\|\hat{\mathbf{f}}_i - \mathbf{C}_j\|_2^2)$ is a Gaussian RBF with a fixed unit scale, whose response decays exponentially with the feature-to-prototype distance. Normalizing these responses across all K source prototypes yields a soft assignment $a_{i,k}$, rather than a hard nearest-prototype assignment. Therefore, $a_{i,k}$ quantifies the relative association of target sample $\mathbf{x}_i$ with the k-th source class prototype.

For an incoming batch of B target samples, we aggregate these sample-wise prototype association scores to estimate the batch-level class tendency as $\pi_k^{\text{batch}} = (1/B) \sum_{i=1}^{B} a_{i,k}$. Thus, π_k^{batch} summarizes the overall tendency of the current target batch toward the k-th class. This estimated tendency is subsequently used to correct the source-induced prediction bias before pseudo-label generation.

3.3 Pseudo-labeling and Tendency-Corrected Feature Adaptation

The estimated batch-level class tendency π_k^{batch} provides target-side information about the class preference of the current batch. Although normalization removes class-dependent vector norm variations, the decision boundaries inherited from long-tailed source training may still favor source-majority classes. As a consequence, using the predictions from these prototype-based logits without correction for pseudo-labeling would propagate the remaining bias into adaptation. We thus use the estimated class tendency to correct the prototype-based logits before pseudo-label generation.

For target sample $\mathbf{x}_i$, we first compute its classification logit with respect to the normalized source prototypes as $z_{i,k} = \hat{\mathbf{f}}_i^{\top} \mathbf{C}_k$. We then incorporate the estimated batch-level class tendency through logit adjustment, $\tilde{z}_{i,k} = z_{i,k} - \alpha \log(\pi_k^{\text{batch}} + \epsilon)$, where α controls the adjustment strength and ϵ is a small constant for numerical stability. The adjusted prediction is obtained as $q_{i,k} = \exp(\tilde{z}_{i,k}) / \sum_{j=1}^{K} \exp(\tilde{z}_{i,j})$. Accordingly, the pseudo-label and its confidence are defined as $\hat{y}_i = \arg\max_k q_{i,k}$ and $\gamma_i = \max_k q_{i,k}$, respectively. Only samples satisfying $\gamma_i > \tau$ are selected for the cross-entropy objective. Note that $a_{i,k}$ in Sec. 3.2 and $q_{i,k}$ serve different purposes: $a_{i,k}$ measures the geometric association used to estimate the batch-level class tendency, whereas $q_{i,k}$ is the tendency-corrected prediction used for pseudo-label generation.

The obtained pseudo-labels then provide supervision for updating the batch normalization layers through the cross-entropy and prototype-alignment objectives. Let $\mathcal{M} = \{i \mid \gamma_i > \tau\}$ denote the set of high-confidence target samples. We compute the cross-entropy loss using the unadjusted logits as $\mathcal{L}_{\text{CE}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \text{CE}\,(z_{i,\hat{y}_i}, \hat{y}_i)$. In addition, we introduce a prototype-alignment objective over the target batch, $\mathcal{L}_{\text{align}} = \frac{1}{B} \sum_{i=1}^{B} [1 - \cos\,(\mathbf{f}_i, \mathbf{C}_{\hat{y}_i})]$, which encourages each target feature to align with the fixed source prototype indicated by its corrected pseudo-label via cosine similarity. The overall adaptation objective is therefore $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}}$, in which $\mathcal{L}_{\text{align}}$ encourages target features to align with their corresponding source prototypes.

The estimated pseudo-label does not directly modify the current target feature. Instead, the resulting supervision updates the trainable parameters of the batch normalization layers, which subsequently changes the feature representations of incoming target samples. In this sense, the estimated class tendency first reduces the source-induced bias in deciding which prototype a target sample should follow, while the adaptation objective converts this assignment into normalization updates that realize prototype-driven feature adaptation.

3.4 Warm-up Strategy

We empirically observe that hyperspherical feature normalization improves the subsequent adaptation but slows down the initial adaptation process. We therefore introduce a short warm-up strategy over the first T_w batches, allowing the model to perform an initial adaptation before starting the complete HyPFA-TTA procedure.

The warm-up stage mainly differs from the complete procedure in two aspects. First, target features retain their original magnitudes **without normalization**, while the source prototypes remain normalized as defined in Sec. 3.1. Accordingly, the classification logits during warm-up are computed as $z_{i,k} = \mathbf{f}_i^\top \mathbf{C}_k$. Second, class-tendency estimation and the corresponding logit adjustment are temporarily disabled. The pseudo-label $\hat{y}_i$ and its confidence score γ_i are instead obtained directly from the distance-based prototype assignments computed using the unnormalized target features. That is, $\hat{y}_i = \arg\max_k p_{i,k}$ and $\gamma_i = \max_k p_{i,k}$, where $p_{i,k} = \frac{\exp(-d_{i,k})}{\sum_{j=1}^{K} \exp(-d_{i,j})}$ and $d_{i,k} = \|\mathbf{f}_i - \mathbf{C}_k\|_2^2$. The same adaptation objective in Sec. 3.3 is retained to update the parameters of the normalization layers.

After the first T_w target batches, HyPFA-TTA activates hyperspherical feature normalization together with class-tendency estimation and logit adjustment, and follows the complete prototype-driven feature adaptation procedure described in the previous subsections.

4 Experiment Results

4.1 Experiment Setup

• Datasets and Evaluation Protocol. We primarily evaluate HyPFA-TTA on the **ImageNet-LT** → **ImageNet-C** setting as our main public benchmark. ImageNet-LT Liu et al. (2019) provides a long-tailed source distribution over 1000 classes with an imbalance ratio of 256. We use a ResNet-101 He et al. (2016) pretrained on **ImageNet-LT** under the instance-balanced setting Kang et al. (2020) as the source model. **ImageNet-C** Hendrycks & Dietterich (2019) contains 15 corruption types with five severity levels and provides a class-balanced target benchmark for evaluating distribution shifts. This setting therefore allows us to examine TTA under long-tailed source pretraining without confounding the evaluation with a long-tailed target distribution. Unless otherwise specified, we report classification accuracy under corruption severity 5.

We further evaluate HyPFA-TTA on **Whitebait-31**→**Whitebait-31Ex**, a real-world highly-imbalanced classification setting[1]. **Whitebait-31** contains training 28,180 images from 31 juvenile and larval fish classes and is used for source pretraining, while **Whitebait-31Ex** contains 12,296 images from the same 31 classes and serves as the target test set. The two datasets exhibit different class distributions and imaging conditions, providing a real-world setting involving both source imbalance and domain shift. The performance of the pretrained source models on their original source test sets is provided in Appendix A.2.

• Compared state-of-the-art methods. We compare HyPFA-TTA with state-of-the-art TTA methods, including TENT Wang et al. (2021), SAR Niu et al. (2023), RoTTA Yuan et al. (2023), DeYO Lee et al. (2024), and ROID Marsden et al. (2024). The performance of the pretrained model, denoted as **Source**, is also reported as a baseline. For Whitebait-31 → Whitebait-31Ex, we additionally include LAME Boudiaf et al. (2022) and Tsallis entropy

[1] **Whitebait-31** and **Whitebait-31Ex** are private datasets available at `https://osf.io/jebs9/overview?view_only=8d35e9b8c2ec4e2ba35ae417baa54e0d`. Dataset properties are provided in Appendix A.4.

Table 1: Performance comparison on ImageNet-C using models pretrained on ImageNet-LT (Batch size=32). All results are classification accuracy (%) under corruption severity 5.

	Noise			Blur				Weather				Digital			
Method	Gauss.	Shot	Imp.	Defoc.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG
Source	0.60	0.94	0.54	6.74	6.21	6.34	10.21	11.20	10.68	9.07	32.14	3.41	14.13	13.12	16.34
TENT	2.09	2.71	2.05	2.37	1.94	3.92	5.22	4.06	2.69	7.00	10.65	1.15	6.55	7.42	5.94
SAR	2.24	3.15	2.94	0.92	1.19	4.77	15.54	13.69	6.11	25.49	32.63	1.97	24.52	27.48	22.44
RoTTA	10.10	11.29	9.00	9.31	9.57	14.97	20.16	17.34	14.74	22.01	31.22	10.38	24.12	25.96	23.25
DeYO	0.35	0.34	0.29	0.26	0.20	0.71	1.69	0.99	0.53	2.61	6.47	0.26	1.61	2.97	1.89
ROID	8.95	10.06	8.80	8.39	8.28	15.18	19.48	17.84	12.45	23.01	29.25	0.93	22.73	24.86	23.06
Ours	**13.76**	**14.77**	**13.04**	**14.43**	**13.53**	**20.67**	**23.62**	**22.03**	**17.80**	**27.34**	**34.11**	**15.94**	**27.99**	**29.37**	**27.28**

adaptation Wu et al. (2026) to provide a broader comparison under the real-world highly-imbalanced setting.

• Implementation details. We use ResNet-101 as the backbone for both **ImageNet-LT→ImageNet-C** and **Whitebait-31→Whitebait-31Ex**. Following the online TTA protocol, each incoming unlabeled target mini-batch is observed once and used for one model update, without access to source samples. We use SGD with a learning rate of 1×10^{-3} and momentum of 0.9. The batch sizes are set to 32 for ImageNet-C and 16 for Whitebait-31Ex. For HyPFA-TTA, we set the confidence threshold to $\tau = 0.5$, the logit-adjustment strength to $\alpha = 0.1$, the prototype-alignment weight to $\lambda_{\text{align}} = 1$, and the warm-up length to $T_w = 10$ batches. Additional experiments on CIFAR-100-LT → CIFAR-100-C Hendrycks & Dietterich (2019) are provided in the Appendix A.5.

4.2 Performance Evaluation on ImageNet-C

We evaluate HyPFA-TTA on ImageNet-C using the ResNet-101 model pretrained on ImageNet-LT. Table 1 reports the classification accuracy across all 15 corruption types at severity level 5. The source model is pretrained on a highly-imbalanced distribution, whereas the ImageNet-C target set is balanced. HyPFA-TTA achieves the best average accuracy among all compared methods. The combination of long-tailed source pretraining and target distribution shift substantially degrades the source model's recognition performance. Existing TTA methods improve the source model, but their performance remains limited when adaptation is driven by predictions inherited from the long-tailed source model. In contrast, HyPFA-TTA consistently improves performance across different corruption categories and achieves the highest accuracy on all 15 corruption types.

These results demonstrate the importance of correcting source-induced prediction bias in TTA with long-tailed pretrained models and suggest that HyPFA-TTA reduces its influence during pseudo-label generation.

4.3 Performance across Class Groups on ImageNet-C

To further examine whether HyPFA-TTA remains effective across classes with different source training frequencies, we evaluate its performance across different class groups of ImageNet-LT. According to the number of source training samples, the 1000 classes are divided into four groups: Head classes ($n_c > 10^3$), Many-shot classes ($100 < n_c \leq 10^3$), Medium-shot classes ($20 < n_c \leq 100$), and Few-shot classes ($n_c \leq 20$). We report the results on four representative ImageNet-C corruptions, including Gaussian noise, defocus blur, contrast, and elastic transform. As shown in Table 2, HyPFA-TTA achieves the best performance across all four class groups under each evaluated corruption. For Gaussian noise, HyPFA-TTA achieves 50.40%, 24.15%, 8.82%, and 1.36% accuracy on the Head, Many-shot, Medium-shot, and Few-shot groups, respectively. Under contrast corruption, the corresponding accuracies are 57.60%, 27.84%, 10.22%, and 1.92%. Similar behavior is observed under defocus blur and elastic transform. These results show that the effectiveness of HyPFA-TTA extends across all four class groups, although the absolute accuracy remains strongly affected by the number of source training samples. In particular, the improvements on the Many-shot, Medium-shot, and Few-shot classes indicate that the benefit of HyPFA-TTA is not limited to the Head classes, supporting its effectiveness under different degrees of

Table 2: Performance across different source class groups on representative ImageNet-C corruptions. For ImageNet-LT, classes are grouped according to the number of source training samples n_c: Head classes (n_c>10^3), Many-shot classes (10^2< n_c≤10^3), Medium-shot classes (20<n_c≤10^2), and Few-shot classes (n_c≤20). All results are classification accuracy (%).

Corrupt.	Method	All	Head	Many	Med.	Few	Corrupt.	Method	All	Head	Many	Med.	Few
	Source	0.60	0.80	0.92	0.43	0.29		Source	6.74	25.20	12.49	3.87	0.37
	TENT	2.09	28.80	3.73	1.07	0.16		TENT	2.37	24.00	4.49	1.14	0.08
Gaussian	RoTTA	10.10	42.40	18.46	5.87	0.82	Defocus	RoTTA	9.31	42.00	17.13	5.30	0.74
(noise)	ROID	8.95	47.20	17.11	4.53	0.60	(blur)	ROID	8.39	41.20	15.80	4.46	0.63
	DeYO	0.35	5.60	0.65	0.14	0.05		DeYO	0.26	6.80	0.55	0.04	0.01
	Ours	**13.76**	**50.40**	**24.15**	**8.82**	**1.36**		**Ours**	**14.48**	**46.40**	**25.07**	**9.49**	**1.86**
	Source	3.41	20.80	6.72	1.58	0.12		Source	14.13	47.20	22.86	10.32	2.51
	TENT	1.51	20.80	2.63	0.82	0.15		TENT	6.55	46.00	11.16	4.22	0.7
Contrast	RoTTA	10.38	44.80	18.77	6.16	0.95	Elastic	RoTTA	24.12	76.00	39.88	17.28	3.32
(digital)	ROID	0.93	4.00	1.76	0.49	0.07	(digital)	ROID	22.73	68.40	38.52	15.59	2.99
	DeYO	0.26	8.00	0.46	0.09	0.00		DeYO	1.61	11.60	2.71	1.02	0.21
	Ours	**15.94**	**57.60**	**27.84**	**10.22**	**1.92**		**Ours**	**27.99**	**80.40**	**45.38**	**20.77**	**4.12**

Table 3: Performance comparison and ablation studies of HyPFA-TTA on Whitebait-31 → Whitebait-31Ex. Classes are grouped according to the number of source training samples: Head classes (n_c>10^3), Many-shot classes (100<n_c≤10^3), Medium-shot classes (20<n_c≤100), and Few-shot classes (n_c≤20). All results are classification accuracy (%). In the ablation table, *logit correction* is denoted as LC, *vector normalization* as Nm, *prototype update* as PU, and *warm-up* as Wrm.

(A) Performance comparison

Method	All	Head	Many	Med.	Few
Source	25.73	44.51	11.26	6.60	0.00
TENT	36.28	74.51	0.82	2.67	0.00
SAR	29.61	57.76	3.14	6.87	0.00
RoTTA	28.16	57.40	1.64	1.42	0.00
LAME	25.84	47.87	7.53	4.20	0.00
DeYO	17.82	36.94	0.11	0.82	0.00
ROID	14.77	30.88	0.00	0.00	0.00
Tsallis	30.46	56.00	7.66	8.83	0.00
Ours	**42.94**	**70.46**	**19.87**	**20.77**	**0.50**

(B) Component ablation

$\mathcal{L}_{\text{CE}}$	$\mathcal{L}_{\text{align}}$	LC	Nm	PU	Wrm	All	Head	Many	Med.	Few
✓	–	–	✓	–	–	40.26	68.64	14.96	18.70	0.63
✓	✓	–	✓	–	–	40.26	79.00	5.55	5.18	0.00
✓	–	✓	✓	–	–	29.38	53.69	7.74	8.72	0.38
✓	✓	✓	✓	–	–	*41.92*	*67.07*	*22.57*	*19.14*	*0.63*
✓	✓	✓	–	–	–	40.53	77.41	2.67	17.99	0.00
✓	✓	–	✓	✓	–	39.46	67.02	14.85	18.81	0.50
✓	✓	✓	✓	✓	–	41.09	65.39	22.57	18.92	0.88
Our default model										
✓	✓	✓	✓	–	✓	**42.94**	**70.46**	**19.87**	**20.77**	**0.50**

source class imbalance. The adaptation curves for the overall performance and different class groups are further provided in Appendix Sec. A.6.

4.4 Results on the Real-World Highly-imbalanced Whitebait Dataset

We next compare HyPFA-TTA with existing TTA methods on Whitebait-31→Whitebait-31Ex. As shown in Table 3(a), a clear failure mode can be observed among existing TTA methods. Although several methods improve the overall accuracy over the source model, their gains are largely concentrated in the head classes, while the performance of non-head classes is substantially degraded. For example, although TENT achieves a considerable improvement (25.73%→36.28%) in overall accuracy, it favors the head classes (44.51%→74.51%) while severely degrading the recognition of other classes (11.26%→0.82%, and 6.60%→2.67%). Similar results are observed for other TTA methods. These results indicate that conventional TTA may amplify the class bias inherited from highly-imbalanced source pretraining, even when adaptation improves the overall performance.

In contrast, our HyPFA-TTA achieves the highest overall accuracy of 42.94%, while simultaneously improving the accuracies of many-shot classes and medium-shot classes over the source model (*i.e.*, 11.26%→19.87% and 6.60%→20.77%, respectively). These results indicate that the performance gain of HyPFA-TTA is not achieved solely through improved recognition of the head classes. Instead, the improvements across the head, many-shot, and medium-shot classes support its effectiveness in reducing the influence of source-majority bias during target adaptation, with or without the warm-up stage (as shown in Table 3(b)).

Table 4: Performance comparison on ImageNet-C with models pretrained on ImageNet-LT under different test batch sizes. All results are under corruption severity 5.

(A) Batch size = 16 (values here denote classification accuracy.)

Method	Noise			Blur				Weather				Digital			
	Gauss.	Shot	Imp.	Defoc.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG
Source	0.60	0.94	0.54	6.74	6.21	6.34	10.21	11.20	10.68	9.07	32.14	3.41	14.13	13.12	16.34
TENT	0.72	0.85	0.64	0.95	0.83	1.32	1.71	1.54	1.00	2.28	3.59	0.51	2.33	2.72	1.96
SAR	6.58	4.45	4.90	3.60	2.99	2.07	4.50	4.22	4.59	9.13	14.30	2.23	6.65	8.69	4.06
RoTTA	10.10	10.93	8.89	9.42	9.24	14.86	20.09	17.34	**14.98**	21.62	**31.46**	**10.38**	24.38	25.68	23.28
DeYO	0.15	0.13	0.12	0.12	0.15	0.18	0.16	0.23	0.14	0.29	0.70	0.12	0.44	0.53	0.28
ROID	7.84	8.68	7.43	7.62	7.09	12.18	16.73	14.85	11.42	20.07	26.42	0.89	19.45	21.93	20.59
Ours	**11.91**	**12.75**	**11.36**	**12.34**	**11.74**	**17.74**	**20.21**	**19.61**	14.36	**24.33**	30.59	6.96	**24.96**	**26.39**	**23.93**

(B) Batch size = 64 (values here denote classification accuracy.)

Method	Noise			Blur				Weather				Digital			
	Gauss.	Shot	Imp.	Defoc.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG
Source	0.60	0.94	0.54	6.74	6.21	6.34	10.21	11.20	10.68	9.07	32.14	3.41	14.13	13.12	16.34
TENT	4.36	5.49	3.92	5.61	4.62	9.28	11.68	9.87	6.01	15.68	25.36	3.09	16.12	18.32	15.43
SAR	9.95	7.97	6.86	4.51	4.53	**23.26**	**25.74**	**24.11**	**19.09**	**29.58**	**35.57**	9.76	**30.03**	**31.84**	**29.03**
RoTTA	10.09	11.46	8.92	9.30	9.55	14.82	20.32	17.39	14.26	21.95	31.44	10.22	23.75	25.40	23.43
DeYO	0.88	1.27	1.35	1.57	0.63	2.99	5.17	4.50	3.18	5.59	12.30	0.79	7.56	10.19	7.50
ROID	7.56	9.49	7.38	5.91	5.05	15.34	19.79	18.03	9.97	23.78	29.70	1.10	22.75	25.48	23.59
Ours	**14.22**	**15.53**	**13.41**	**14.45**	**13.84**	20.92	24.83	22.37	19.05	28.31	35.53	**16.93**	28.98	30.39	27.88

4.5 Ablation Study

We further analyze the contributions of the main components of HyPFA-TTA on Whitebait-31→Whitebait-31Ex. As shown in Table 3(b), using $\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{align}}$ alone results in a strong bias toward the head classes, indicating that prototype alignment without bias correction can still reinforce the majority-class tendency inherited from source pretraining. Introducing logit correction (LC) substantially improves the recognition of the many-shot and medium-shot classes and yields a more balanced adaptation across different class groups. In addition, updating the source prototypes (ProtoUpdate, *i.e.*, **PU**, in Table 3) does not provide further improvement, supporting our design of fixed prototypes during adaptation. Table 3(b) further examines the effect of feature normalization and the warm-up strategy. Without feature normalization, the adapted model again becomes strongly biased toward the head classes, while the recognition of the other groups deteriorates substantially. Hyperspherical normalization alleviates this imbalance by removing class-dependent vector norm variations during prototype-based estimation. The warm-up strategy further improves the overall performance, supporting its use before the complete HyPFA-TTA procedure.

We additionally evaluate the sensitivity to the test batch size on ImageNet-C. As shown in Table 4, HyPFA-TTA achieves the best average performance under both batch sizes of 16 and 64, while maintaining strong performance across different corruption types. Moreover, its performance improves as the batch size increases, whereas some other methods show only limited changes. This behavior is consistent with our batch-level class-tendency estimation, since a larger batch provides a more stable estimate of the batch-level class tendency while the method remains effective even with relatively small batches. A summary of batch-size experiments is provided in Appendix A.3.

5 Concluding Remarks

We study test-time adaptation under long-tailed source pretraining, where conventional TTA may amplify the class bias inherited from the source model. To address this issue, we formulate TTA as prototype-driven feature adaptation and propose HyPFA-TTA, which uses normalized source classifier weights as fixed prototypes and corrects source-biased predictions using the estimated class tendency of each target batch. Experiments on ImageNet-C and the real-world Whitebait-31/Whitebait-31Ex dataset show consistent improvements across different class groups, demonstrating that the benefits of HyPFA-TTA extend beyond the head classes to classes with fewer source training samples.

AI Use Disclosure

ChatGPT was used to assist with revising and polishing the manuscript, including improvements to language, grammar and typo checking, and table formatting and layout.

References

S. Alshammari, Y.-X. Wang, D. Ramanan, and S. Kong. Long-tailed recognition via weight balancing. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 6887–6897, 2022.

M. Boudiaf, R. Mueller, I. Ben Ayed, and L. Bertinetto. Parameter-free online test-time adaptation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 8344–8353, 2022.

D. Chen, D. Wang, T. Darrell, and S. Ebrahimi. Contrastive test-time adaptation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 295–305, 2022.

Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 9268–9277, 2019.

H. Dang, T. Huu, S. Osher, H. Tran, N. Ho, and T. Nguyen. Neural collapse in deep linear networks: From balanced to imbalanced data. In *Int. Conf. Mach. Learn.*, volume 202, pp. 6873–6947, 2023.

J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 4690–4699, 2019.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, 2016.

D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Int. Conf. Learn. Represent.*, 2019.

Y. Iwasawa and Y. Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *Adv. Neural Inf. Process. Syst.*, volume 34, pp. 2427–2440, 2021.

B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Int. Conf. Learn. Represent.*, 2020.

J. Lee, D. Jung, S. Lee, J. Park, J. Shin, U. Hwang, and S. Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *Int. Conf. Learn. Represent.*, 2024.

Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. Revisiting batch normalization for practical domain adaptation. In *Int. Conf. Learn. Represent.*, 2017.

J. Liang, D. Hu, and J. Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Int. Conf. Mach. Learn.*, pp. 6028–6039, 2020.

J. Liang, R. He, and T. Tan. A comprehensive survey on test-time adaptation under distribution shifts. *Int. J. Comput. Vis.*, 133(1):31–64, 2025. doi: 10.1007/s11263-024-02181-w.

W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 212–220, 2017.

Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. Yu. Large-scale long-tailed recognition in an open world. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 2537–2546, 2019.

R. Marsden, M. Dobler, and B. Yang. Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, pp. 2543–2553, 2024.

A. Menon, S. Jayasumana, A. Rawat, H. Jain, A. Veit, and S. Kumar. Long-tail learning via logit adjustment. In *Int. Conf. Learn. Represent.*, 2021.

S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao, and M. Tan. Efficient test-time model adaptation without forgetting. In *Int. Conf. Mach. Learn.*, volume 162, pp. 16888–16905, 2022.

S. Niu, J. Wu, Y. Zhang, Z. Wen, Y. Chen, P. Zhao, and M. Tan. Towards stable test-time adaptation in dynamic wild world. In *Int. Conf. Learn. Represent.*, 2023.

S. Park, S. Yang, J. Choo, and S. Yun. Label shift adapter for test-time adaptation under covariate and label shifts. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 16421–16431, 2023.

D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Tent: Fully test-time adaptation by entropy minimization. In *Int. Conf. Learn. Represent.*, 2021.

H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5265–5274, 2018.

Q. Wang, O. Fink, L. Van Gool, and D. Dai. Continual test-time domain adaptation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 7201–7211, 2022.

X. Wu, D. Jiang, F. Yu, Y. Tian, J. Tang, Q.-G. Chen, Y. Yang, and J. Lu. Adaptive debiasing tsallis entropy for test-time adaptation. In *Int. Conf. Learn. Represent.*, 2026.

L. Yuan, B. Xie, and S. Li. Robust test-time adaptation in dynamic scenarios. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 15922–15932, 2023.

M. Zhang, S. Levine, and C. Finn. Memo: Test time robustness via adaptation and augmentation. In *Adv. Neural Inf. Process. Syst.*, volume 35, 2022.

A Appendix

Overview. This appendix provides supplementary algorithmic details, additional analyses, and experimental results that complement the main paper. Fig. 2 below demonstrates how ImageNet-C target features are adapted through the proposed HyPFA-TTA.

- **Sec. A.1 (pages 13–14)** presents the complete pseudo-code and flow-chart of the proposed HyPFA-TTA, corresponding to the adaptation procedure described in Secs. 3.2–3.4.
- **Sec. A.2 (page 15)** reports the performance of the pretrained source models on their original source test sets, complementing the experimental setup in Sec. 4.1.
- **Sec. A.3 (page 16)** summarizes the sensitivity of HyPFA-TTA and other SOTAs to different test batch sizes, extending the batch-size analysis in Sec. 4.5.
- **Sec. A.4 (pages 17–18)** analyzes the class-dependent vector norm variations under long-tailed pretraining, providing empirical support for the prototype normalization introduced in Sec. 3.1. This subsection also provides an overview of the whitebait specimen image dataset.
- **Sec. A.5 (page 19)** provides additional results on CIFAR-100-LT→CIFAR-100-C, extending the evaluation settings described in Sec. 4.1.
- **Sec. A.6 (pages 20–24)** presents adaptation curves under different corruption types to complement the numerical evaluation in Secs. 4.2–4.3.

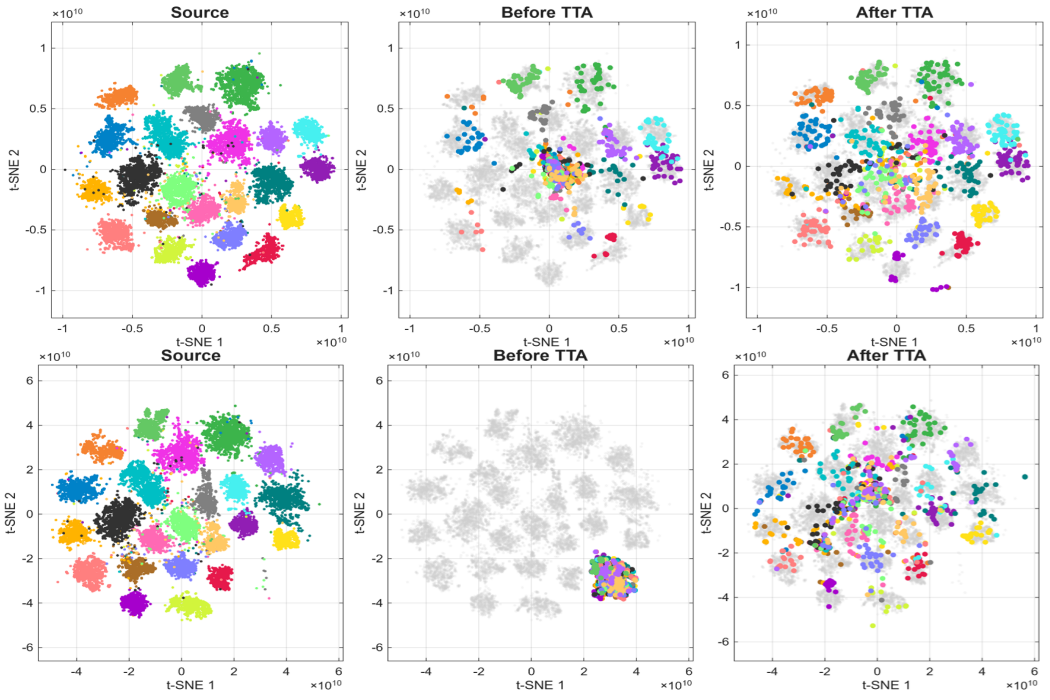


Figure 2: **t-SNE visualization of source and target features for ImageNet-LT→ImageNet-C under *elastic transform* (top row) and *additive Gaussian noise* (bottom row) at severity 5.** Only classes with more than 500 ImageNet-LT training samples are shown. Left: source features extracted from the ImageNet-LT training subset. Middle and Right: target features before and after our proposed HyPFA-TTA, respectively; gray points represent source features, and colors indicate different target classes. All panels share the same t-SNE coordinate space. These two rows correspond to the experiments reported in the lower-right/upper-left parts of Table 2.

A.1 Algorithm & Flow-Chart

Algorithm 1 HyPFA-TTA

Require: Target stream $\mathcal{D}_t$, pretrained model F and its feature extractor Φ, classifier weight matrix $\mathbf{W}$, warm-up iteration T_w, confidence threshold τ, logit-adjustment strength α, prototype-alignment weight λ_{align}, batch size B.
Ensure: Adapted model F^*.

Normalize source prototypes: $\mathbf{C} \leftarrow \text{Normalize}(\mathbf{W})$
for each target batch $\mathcal{X}_t = \{x_i\}_{i=1}^{B}$ at iteration t **do**
  Extract target features: $\mathbf{f}_i \leftarrow \Phi(x_i), \quad i = 1, \dots, B$
  if $t < T_w$ **then**
    // Warm-up (Section 3.4)
    Compute classification logits: $z_{i,k} \leftarrow \mathbf{f}_i^\top \mathbf{C}_k$
    Compute feature–prototype distances: $d_{i,k} \leftarrow \|\mathbf{f}_i - \mathbf{C}_k\|_2^2$
    Compute prototype assignments: $p_{i,k} \leftarrow \frac{\exp(-d_{i,k})}{\sum_{j=1}^{K} \exp(-d_{i,j})}$
    Obtain pseudo-labels and confidence: $\hat{y}_i \leftarrow \arg\max_k p_{i,k}$ and $\gamma_i \leftarrow \max_k p_{i,k}$
  else
    // Complete HyPFA-TTA (Sections 3.2 and 3.3)
    Project target features onto the unit hypersphere via normalization: $\hat{\mathbf{f}}_i \leftarrow \mathbf{f}_i / \|\mathbf{f}_i\|_2$
    Compute classification logits with respect to normalized prototypes: $z_{i,k} \leftarrow \hat{\mathbf{f}}_i^\top \mathbf{C}_k$
    Compute hyperspherical feature–prototype distances: $d_{i,k} \leftarrow \|\hat{\mathbf{f}}_i - \mathbf{C}_k\|_2^2$
    Compute prototype associations via Gaussian RBF: $a_{i,k} \leftarrow \frac{\exp(-d_{i,k})}{\sum_{j=1}^{K} \exp(-d_{i,j})}$
    Estimate batch-level class tendency: $\pi_k^{\text{batch}} \leftarrow \frac{1}{B} \sum_{i=1}^{B} a_{i,k}$
    Perform logit adjustment: $\tilde{z}_{i,k} \leftarrow z_{i,k} - \alpha \log(\pi_k^{\text{batch}} + \epsilon)$
    Compute tendency-corrected predictions: $q_{i,k} \leftarrow \frac{\exp(\tilde{z}_{i,k})}{\sum_{j=1}^{K} \exp(\tilde{z}_{i,j})}$
    Obtain pseudo-labels and confidence: $\hat{y}_i \leftarrow \arg\max_k q_{i,k}$ and $\gamma_i \leftarrow \max_k q_{i,k}$
  end if
  Select high-confidence samples: $\mathcal{M} \leftarrow \{i : \gamma_i > \tau\}$
  Compute cross-entropy using unadjusted logits: $\mathcal{L}_{\text{CE}} \leftarrow \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \text{CE}(z_{i,\hat{y}_i}, \hat{y}_i)$
  Compute prototype alignment loss: $\mathcal{L}_{\text{align}} \leftarrow \frac{1}{B} \sum_{i=1}^{B} [1 - \cos(\mathbf{f}_i, \mathbf{C}_{\hat{y}_i})]$
  Compute adaptation objective: $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{CE}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}}$
  Update parameters of BN layers by back-propagation
end for
return F^*

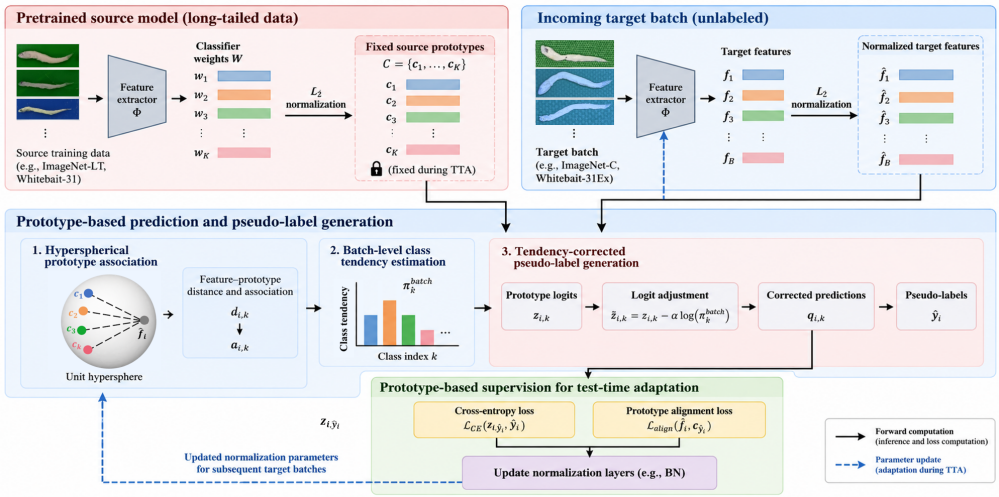


Figure 3: Flow-Chart of HyPFA-TTA (better viewed at 250%). This figure shows only the data flow, and some component modules of this figures (e.g. the unit hypersphere and the histogram) were rendered by ChatGPT.

A.2 Source-Model Performance on the Source Test Sets

Before evaluating test-time adaptation under distribution shifts, we examine the performance of the pretrained source models on their corresponding source test sets. As shown in Tables 5 and 6, the source models achieve overall accuracies of 44.20% on ImageNet-LT and 92.38% on Whitebait-31, respectively. These results confirm that the source models used in our experiments are well trained on their original source distributions.

The subsequent TTA experiments therefore evaluate adaptation from properly trained source models rather than compensating for inadequate source training.

Table 5: Performance of the pretrained source model on the ImageNet-LT test set. Classification accuracy (%) is reported overall and for the Head, Many-shot, Medium-shot, and Few-shot class groups. Classes are grouped according to the number of source training samples: Head classes ($n_c > 10^3$), Many-shot classes ($100 < n_c \leq 10^3$), Medium-shot classes ($20 < n_c \leq 100$), and Few-shot classes ($n_c \leq 20$).

Model	Test Acc.	Head	Many	Medium	Few
Source	44.20	90.40	66.18	37.13	8.14

Table 6: Performance of the pretrained source model on the Whitebait-31 test set. Classification accuracy (%) is reported overall and for the Head, Many-shot, Medium-shot, and Few-shot class groups.

Model	Test Acc.	Head	Many	Medium	Few
Source	92.38	94.14	81.35	68.24	78.79

A.3 Experiments on Batch Sizes

We further investigate the effect of test batch size on the performance of different TTA methods. As shown in Table 7, HyPFA-TTA achieves strong performance even with a relatively small batch size, demonstrating its effectiveness without requiring a large test batch size.

Table 7: Mean classification accuracy (%) on ImageNet-C under different test batch sizes. The mean is computed over all 15 corruption types at severity 5.

Method	BS = 16	BS = 32	BS = 64
Source	10.80	10.80	10.80
TENT	1.58	4.12	10.13
SAR	5.53	14.34	14.99
RoTTA	17.87	17.89	17.86
DeYO	0.25	1.76	4.78
ROID	14.95	15.60	15.46
HyPFA-TTA	**18.08**	**20.43**	**21.51**

A.4 Class-Dependent Vector Norm Variations under Long-Tailed Pretraining and Whitebait Dataset Overview

Previous theoretical analysis of neural collapse under imbalanced training has shown that class imbalance can induce class-dependent vector norm variations in the learned classifiers and feature representations, where their vector norms depend on the number of training samples in the corresponding classes Dang et al. (2023). In particular, classifier weights associated with majority classes tend to have larger norms than those associated with minority classes.

We examine whether such class-dependent vector norm variations also occur in our real-world long-tailed setting. Table 8 reports the prototype norm of each class together with its class ID for the source model trained on Whitebait-31[2]. The average prototype norm decreases from 1.697 for Head classes to 0.992, 0.875, and 0.682 for many-shot, medium-shot, and few-shot classes, respectively. These results show that class-dependent vector norm variations are also present in the source model trained on the real-world Whitebait-31 dataset.

This empirical observation is consistent with the class-dependent vector norm variations described in prior theoretical analysis and provides additional motivation for removing classifier-weight norm differences when constructing the source reference prototypes. By normalizing the source classifier weights, HyPFA-TTA reduces the influence of the class-dependent vector norm variations inherited from long-tailed source training and uses the resulting normalized class directions as fixed reference prototypes.

In addition, Whitebait-31 consists of specimen images of 31 classes of larval and juvenile fishes. Table 9 lists the number of images in each class. For pretraining the source classification model, we randomly split Whitebait-31 into training, validation, and test subsets at a ratio of 4:1:1. The exact data partition used in our experiments is provided with the released dataset.

[2]Our Whitebait-31/Whitebait-31Ex datasets are currently available for download at this anonymous link: `https://osf.io/jebs9/overview?view_only=8d35e9b8c2ec4e2ba35ae417baa54e0d`.

Table 8: Prototype norms under long-tailed pretraining on Whitebait-31. The average prototype norm decreases from the Head to the Few-shot classes, showing class-dependent feature vector norm variations. Within each group, the classes are ordered by the number of samples in descending order. The number of samples for each class is provided in Table 9.

Head Classes ($n_c > 10^3$)

Class ID	Prototype Norm
23	1.787
25	1.796
26	1.552
24	1.653
Avg.:	**1.697**

Many-shot Classes ($100 < n_c \leq 10^3$)

Class ID	Prototype Norm
12	1.042
27	1.070
28	1.086
17	0.963
19	0.870
4	0.926
6	0.986
Avg.:	**0.992**

Medium-shot Classes ($20 < n_c \leq 100$)

Class ID	Prototype Norm
0	0.868
1	0.796
2	0.828
5	0.870
8	0.959
9	0.929
10	0.861
11	0.926
15	0.838
Avg.:	**0.875**

Few-shot Classes ($n_c \leq 20$)

Class ID	Prototype Norm
3	0.662
7	0.656
13	0.688
14	0.657
16	0.703
18	0.660
20	0.644
21	0.703
22	0.702
29	0.662
30	0.768
Avg.:	**0.682**

Table 9: Class distribution of Whitebait-31. The four head classes are the main economically important species.

Class ID	# Img.	Class ID	# Img.	Class ID	# Img.
00	120	11	150	22	20
01	50	12	370	23	10090
02	50	13	10	24	2620
03	10	14	20	25	8270
04	240	15	110	26	4300
05	70	16	20	27	330
06	240	17	190	28	330
07	10	18	10	29	10
08	120	19	160	30	30
09	110	20	10		
10	90	21	20		

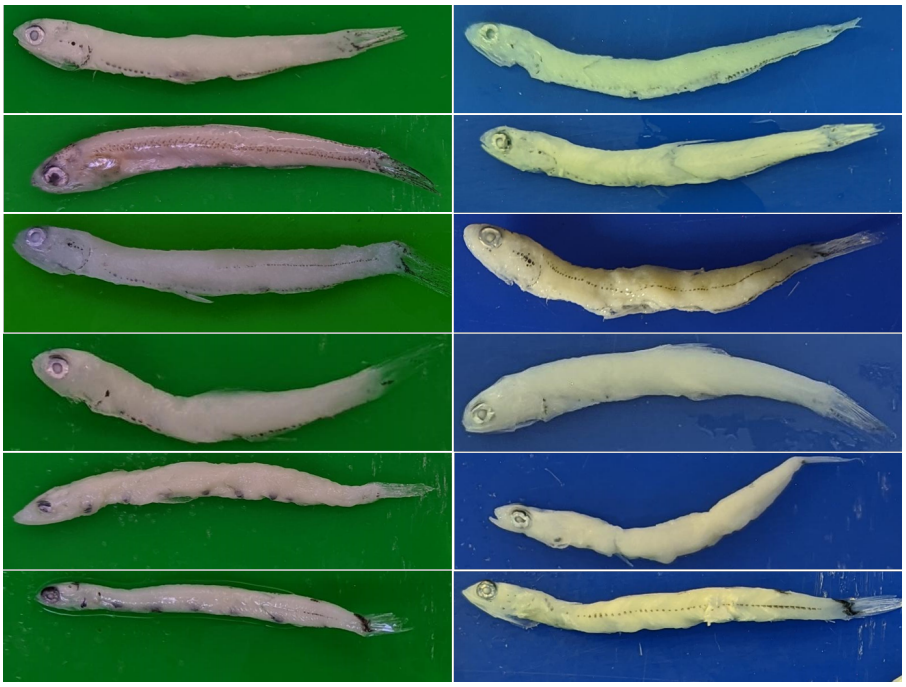


Figure 4: Sample images of Whitebait-31. Whitebait-31 consists of specimen images of *larval and juvenile fishes*. Each row here shows one fish species.

The original data of Whitebait-31 were collected as part of a marine ecological conservation study. These *larval and juvenile fishes* were captured at sea by biologists during research expeditions and subsequently preserved in alcohol as biological specimens. The specimens were then manually classified by a fish identification specialist. For image acquisition, specimens of each class were arranged in groups of ten and photographed to obtain the original images, while specimens were not photographed when fewer than ten were available for a group. Classes with larger numbers of specimens mostly correspond to economically important fish species, whereas classes with fewer specimens mainly consist of bycatch species whose capture should be avoided.

A.5 Experiment Results on CIFAR-100-LT→CIFAR-100-C

Table 10: Performance comparison on CIFAR-100-C using models pretrained on CIFAR-100-LT with an imbalance ratio of $\rho = 100$. Classes are grouped according to the number of source training samples: many-shot classes ($n_c > 100$), medium-shot classes ($20 < n_c \leq 100$), and few-shot classes ($n_c \leq 20$). All results are classification accuracy (%).

Corruption	Method	Total	Many	Med.	Few
Gaussian Noise	Source	16.26	30.31	12.23	4.57
	TENT	30.82	58.29	25.74	4.70
	SAR	30.76	57.40	26.23	4.97
	RoTTA	29.20	56.26	23.43	4.37
	DeYO	30.52	57.60	25.74	4.50
	ROID	32.40	56.40	30.94	6.10
	Ours	30.58	56.86	26.26	4.97
Glass Blur	Source	26.20	44.51	26.17	4.87
	TENT	28.32	52.97	24.20	4.37
	SAR	28.63	52.97	24.89	4.60
	RoTTA	27.29	51.94	22.74	3.83
	DeYO	27.95	53.20	22.89	4.40
	ROID	28.72	50.29	27.40	5.10
	Ours	28.40	52.34	25.00	4.43
Contrast	Source	7.09	13.23	5.26	2.07
	TENT	28.33	53.91	23.11	4.57
	SAR	28.76	54.03	23.60	5.30
	RoTTA	24.46	48.09	18.40	3.97
	DeYO	29.68	56.97	23.63	4.90
	ROID	27.43	49.51	24.91	4.60
	Ours	28.43	53.49	23.26	5.23
Elastic Transform	Source	31.45	54.06	30.89	5.73
	TENT	31.57	58.57	27.00	5.40
	SAR	31.87	58.23	27.91	5.73
	RoTTA	30.79	58.26	25.34	5.10
	DeYO	30.63	58.06	25.14	5.03
	ROID	31.61	54.51	30.94	5.67
	Ours	31.69	57.80	27.74	5.83
Brightness	Source	33.01	56.97	30.83	7.60
	TENT	36.23	65.86	32.03	6.57
	SAR	36.73	66.03	32.89	7.03
	RoTTA	35.49	65.77	30.06	6.50
	DeYO	36.07	65.63	32.11	6.20
	ROID	37.80	63.14	38.57	7.33
	Ours	36.40	65.37	32.71	6.90

A.6 Adaptation curves

While the main paper reports the numerical results, we provide the adaptation curves here to visualize how performance evolves throughout the TTA process. Note that curves shown in this section correspond to the experiments (*i.e.*, ImageNet-LT→ImageNet-C) shown in Table 2.

Fig. 5–Fig. 8 illustrate the evolution of accuracy during TTA under different corruptions, where each target sample is observed only once. The overall results show the cumulative accuracy over the target stream, while the class-group results further show the adaptation behavior across head, many-shot, medium-shot, and few-shot classes.

Overall, HyPFA-TTA consistently improves its accuracy throughout the TTA process, and this trend is observed across different corruptions. In contrast, the competing methods tend to reach saturation earlier or gradually degrade as more target samples are processed. For example, TENT improves rapidly in the early stage but gradually degrades as more target samples are processed. DeYO and ROID exhibit similar decreasing trends, while RoTTA remains relatively stable but reaches a plateau earlier and shows limited improvement in the later stages.

The group-wise results further show that the performance improvement of HyPFA-TTA is not solely driven by the Head classes, but is observed across different class groups. Notably, in the later stages of adaptation, the improvement is mainly contributed by the Many and Medium classes, which continue to exhibit clear upward trends. This suggests that the sustained improvement of HyPFA-TTA is not simply a consequence of adapting to head classes.

These results complement the overall performance reported in the main paper by providing a more detailed view of how adaptation performance evolves over a prolonged target stream. Overall, the results show that the proposed prototype-driven feature adaptation maintains stable and continuous adaptation throughout long-horizon online TTA under models pre-trained with long-tailed source data.

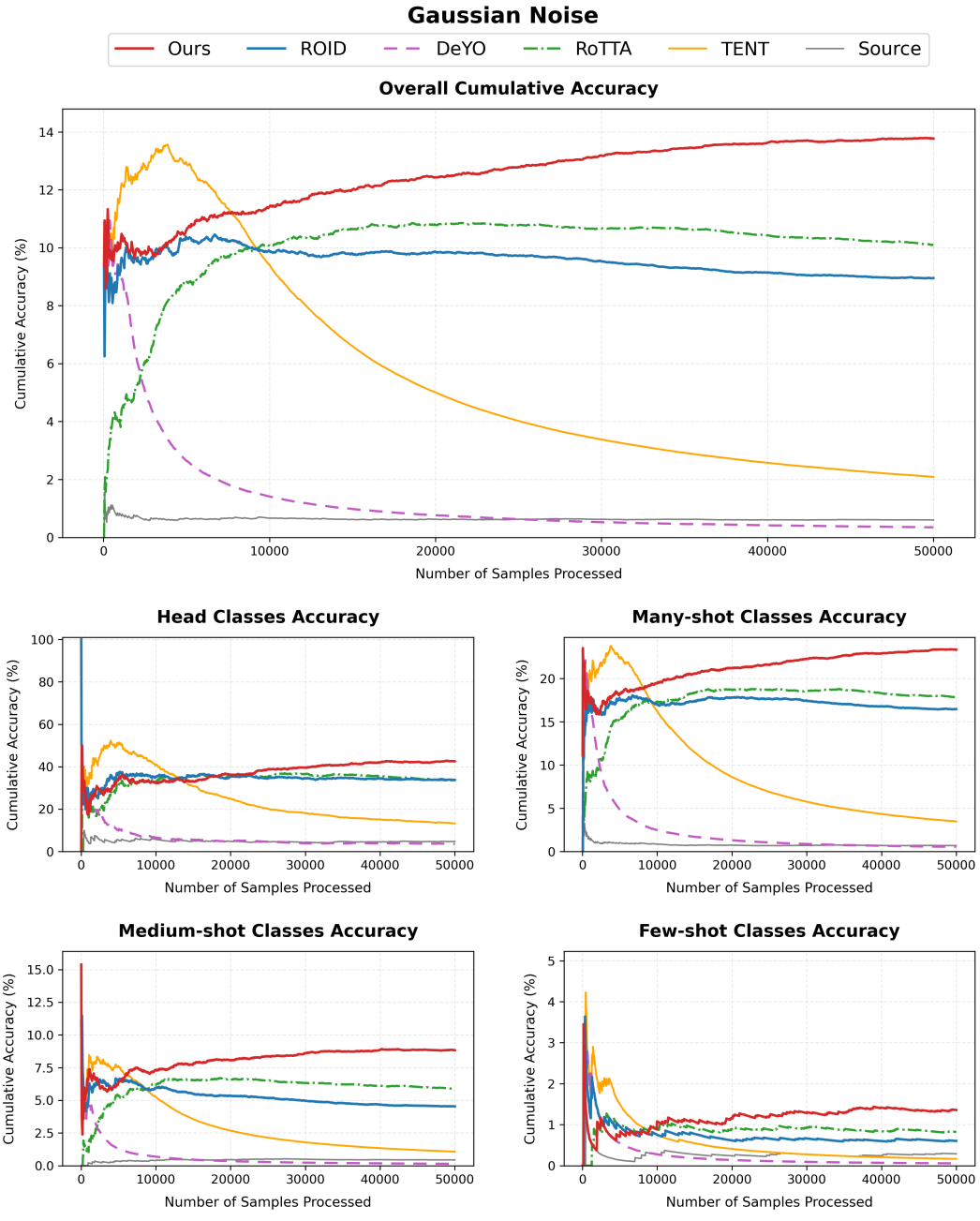


Figure 5: Adaptation behavior of TTA methods under "Gaussian Noise". In ImageNet-C, each corruption type at severity 5 contains 50,000 samples. This figure corresponds to the experiment shown in the *upper-left* part of Table 2.

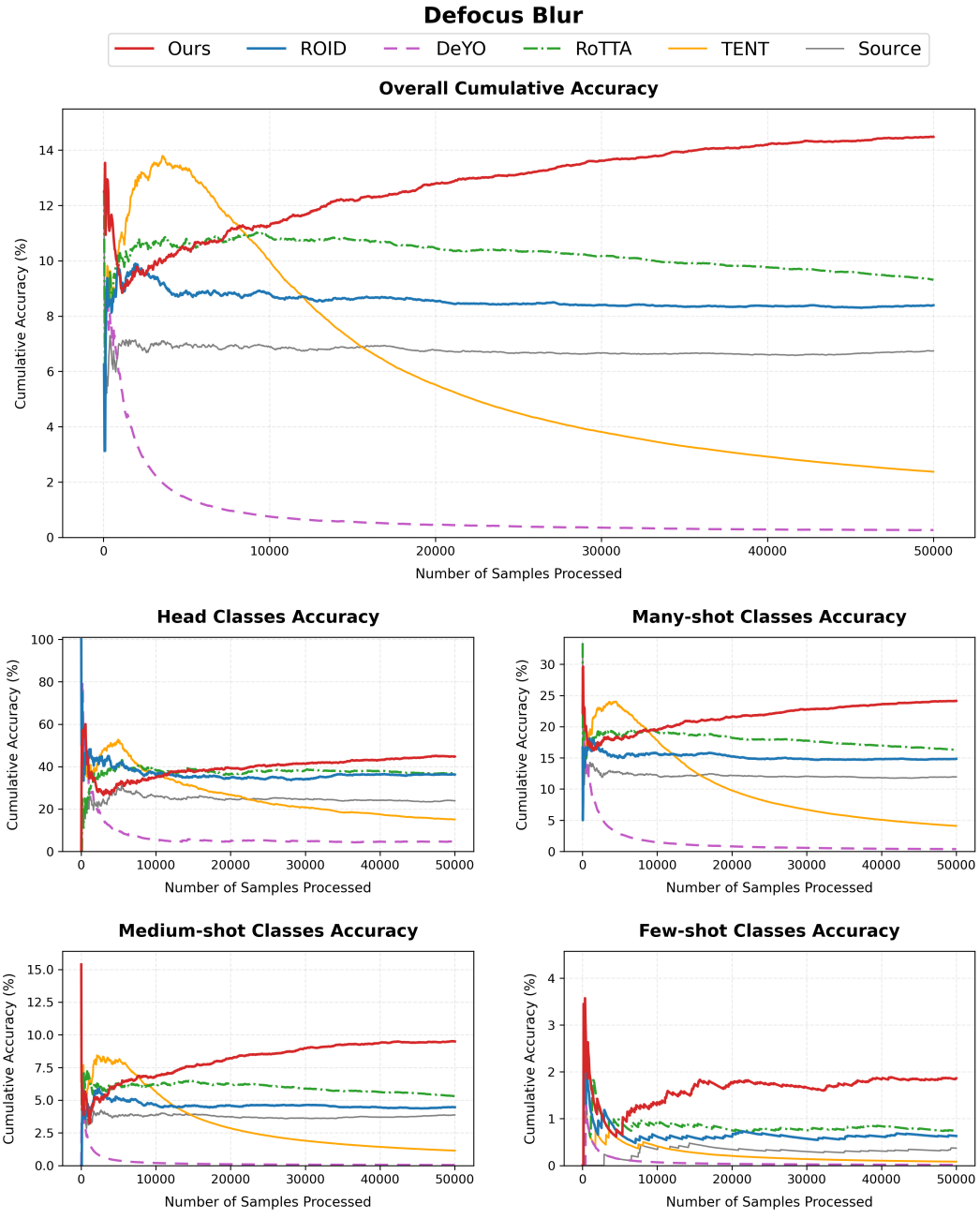


Figure 6: Adaptation behavior of TTA methods under "Defous Blur". In ImageNet-C, each corruption type at severity 5 contains 50,000 samples. This figure corresponds to the experiment shown in the *upper-right* part of Table 2.

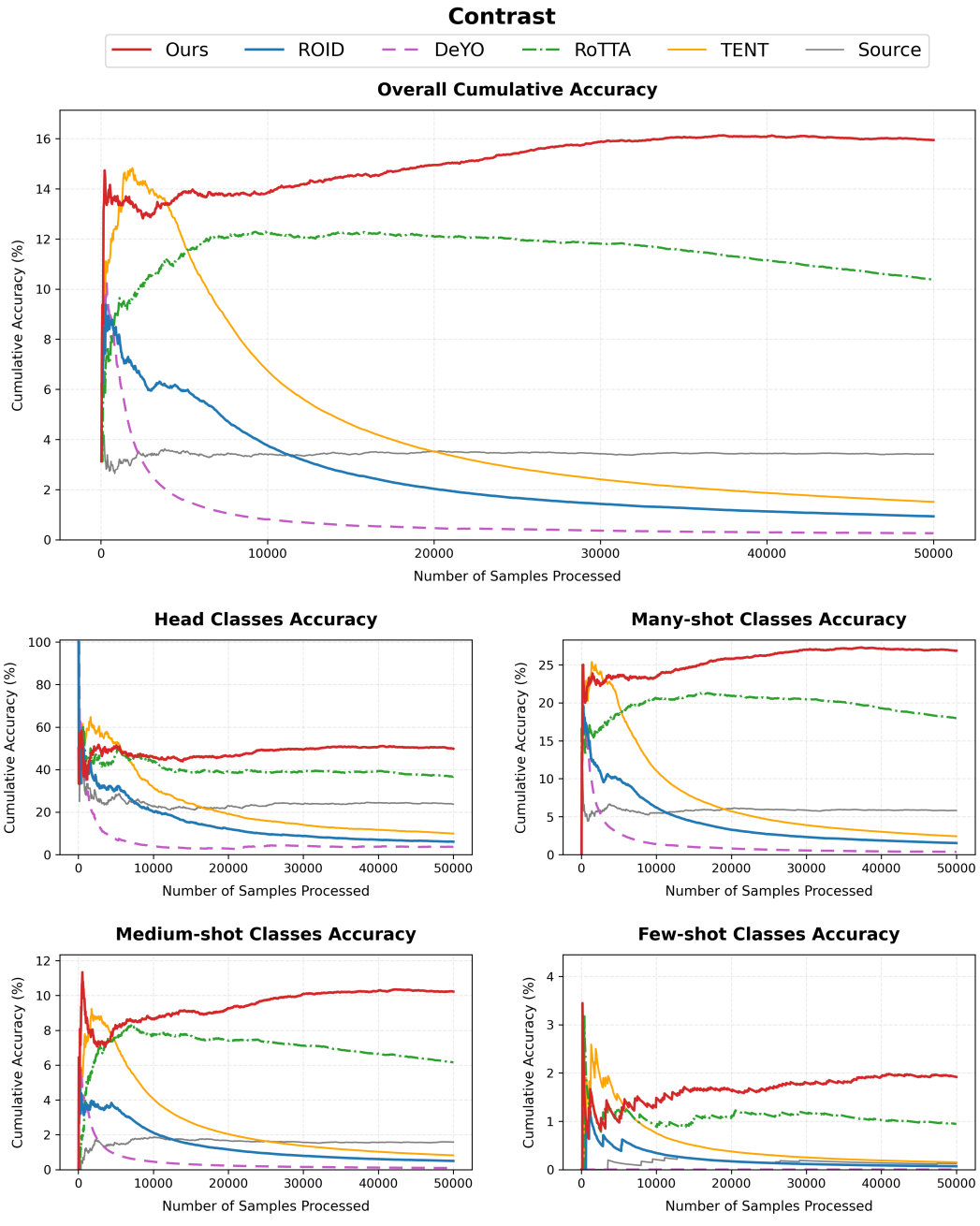


Figure 7: Adaptation behavior of TTA methods under "Contrast". In ImageNet-C, each corruption type at severity 5 contains 50,000 samples. This figure corresponds to the experiment shown in the *lower-left* part of Table 2.

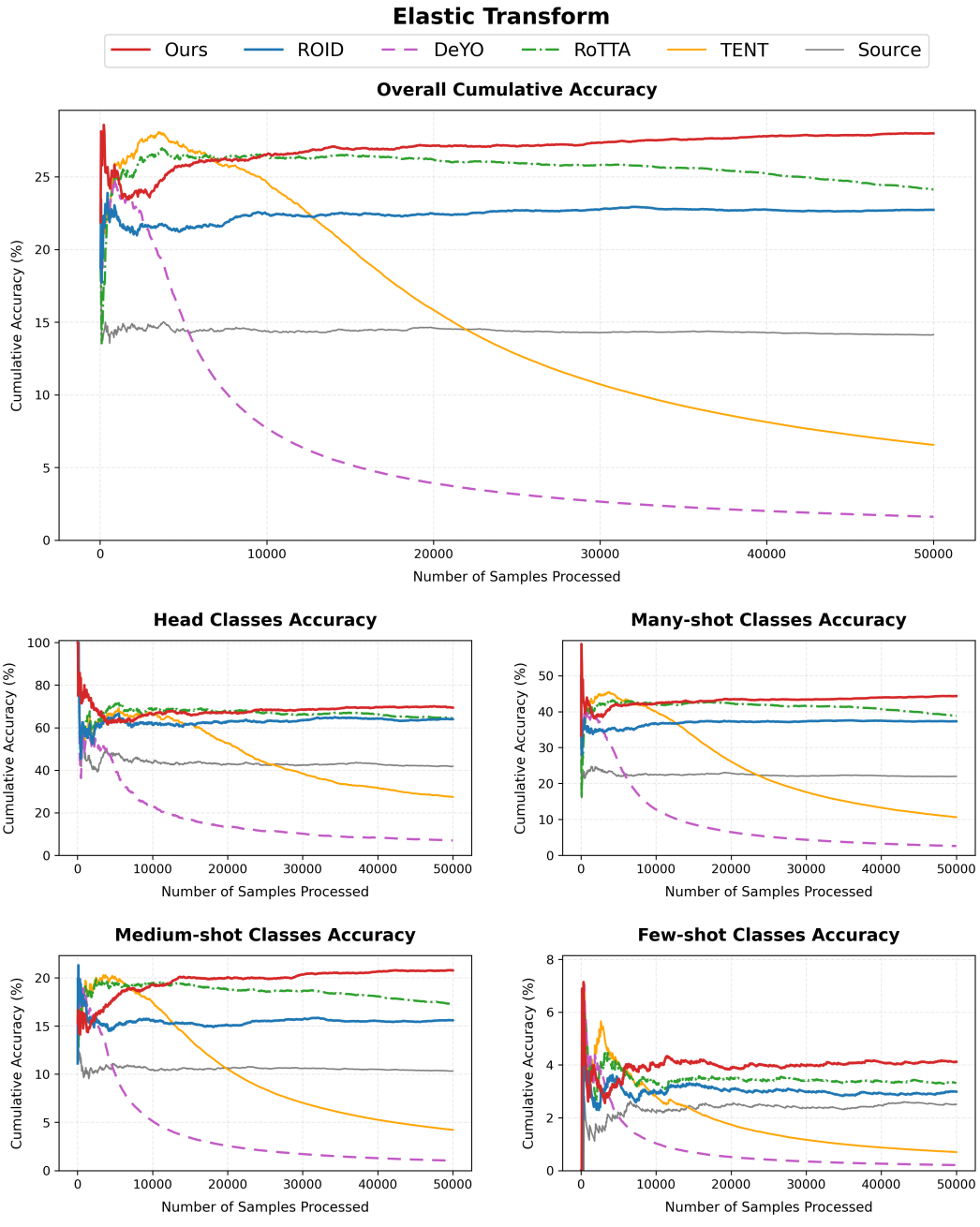


Figure 8: Adaptation behavior of TTA methods under "Elastic Transform". In ImageNet-C, each corruption type at severity 5 contains 50,000 samples. This figure corresponds to the experiment shown in the *lower-right* part of Table 2.