Logit-Adjusted Test-Time Adaptation under Partial Class Imbalance

Thilina Weerasinghe, Ruwan Tennakoon, WeiQin Chuah, Alireza Bab-Hadiashar
RMIT University, Australia
{thilina.weerasinghe,ruwan.tennakoon,wei.qin.chuah,alireza.bab-hadiashar}@rmit.edu.au

Abstract

Test-Time Adaptation (TTA) enables deep neural networks to handle distribution shifts without requiring labels at inference. However, existing methods commonly assume complete class overlap between source and target domains, which rarely holds in practice. We study the challenging setting of ***Partial Class Imbalance****, where the target domain contains only a subset of source classes. We show that entropy minimization–based TTA methods degrade over long test sequences because batch normalization updates bias feature representations toward visible classes, resulting in skewed predictions. To address this, we propose* ***Logit-Adjusted Entropy Minimization****, a simple yet effective strategy that integrates target class priors into the adaptation objective. Our method is model-agnostic and can be seamlessly applied to a wide range of TTA algorithms. Extensive experiments on CIFAR-100-C, ImageNet-C under diverse corruptions and severity levels, and the large-scale DomainNet-126 dataset demonstrate that our method consistently improves adaptation stability and accuracy for both CNNs and Vision Transformers. Compared to strong baselines, our approach reduces overfitting to visible classes and mitigates performance degradation in long-sequence adaptation. Code is available at* https://github.com/thilinawee/latta

1. Introduction

Deep Neural Networks (DNNs) have shown remarkable performance in computer vision tasks. However, their effectiveness often diminishes when test data deviates from the training distribution [18], a common challenge in real-world applications. Test-Time Adaptation (TTA) [21] addresses this limitation by enabling the adaptation of a source-trained model to unlabeled test data during inference.

To be effective in real-world problems, TTA methods must address several critical scenarios. Firstly, TTA methods must consider the possibility that *the unlabeled test data encountered during adaptation may not contain samples from all classes in the original training data*, a scenario we refer to as *partial class imbalance*. Formally, this corresponds to $P^s(y) \neq P^t(y)$, since $\exists\, y \in \mathcal{Y}^s$ such that $y \notin \mathcal{Y}^t$, where $\mathcal{Y}^s$ denotes the set of classes in the source domain and $\mathcal{Y}^t$ the (latent) classes in target domain during adaptation. Notably, in TTA the target domain is unlabeled, making such discrepancies difficult to detect and correct during adaptation. Nevertheless, handling these discrepancies is critical, as previously unseen classes ($\mathcal{Y}^t \setminus \mathcal{Y}^s$) may emerge post-adaptation or after an extended period, undermining models' robustness. Secondly, TTA algorithms cannot rely on prior knowledge of when a distribution shift occurs. This shift can be a gradual and unforeseen process, necessitating TTA algorithms capable of *continuous operation across extended adaptation sequences*. Practical examples of this setting include industrial inspection, where models trained on a diverse set of object types [2] are deployed on production lines that undergo distribution shifts over time. Depending on production schedules, only a limited subset of object types may appear repeatedly during extended operation, resulting in partial class imbalance. Similarly, in wildlife monitoring, camera traps may observe only a small subset of animal classes over days or weeks, despite being trained on data covering a wide range of species [1].

Recent TTA research has investigated various forms of class imbalance in target domains, including online class imbalance [15] and long-tailed class distributions [16, 20], which highlight disparities in class frequencies due to sample counts or temporal order in the test data. However, to the best of our knowledge, *partial class imbalance* remains largely unaddressed. In addition, ensuring stable adaptation over extended periods remains a significant challenge, with only a few explicitly considering long-term adaptation scenarios [15, 22]. The combination of partial class imbalance and prolonged test sequences poses unique challenges that have been largely overlooked by existing approaches.

In this paper, we empirically demonstrate that TTA methods based on entropy minimization suffer from performance degradation when adapting to *long test sequences under partial class imbalance* (see Section 3). To better understand this phenomenon, we theoretically analyse a simplified model. Our analysis reveals that, in the presence of

partial class imbalance, adapting batch normalization parameters (β, γ), which is a common practice in TTA, causes the β parameter to drift and align with the dominant classes observed during adaptation. We refer to this effect as β-*drift* (more details in Sec 3.1). To mitigate this, we introduce a logit adjustment technique that can be seamlessly integrated into existing TTA frameworks. We theoretically show that this adjustment offsets the shift in β, and through extensive experiments across diverse datasets and model architectures, it is shown that our approach consistently improves the performance of entropy minimization-based TTA methods under partial class imbalance. In this work, we make the following key contributions:

1. We study a novel and realistic TTA scenario involving *partial class imbalance* and *extended data sequences*, and demonstrate that commonly used entropy minimization techniques can lead to significant performance degradation in this setting.
2. We provide both theoretical analysis and empirical evidence to uncover the root causes of this degradation, highlighting the role of batch normalization parameter drift under class imbalance.
3. We propose a novel logit adjustment strategy that effectively mitigates performance decline in TTA settings characterized by long adaptation periods and partial class imbalance, while being easily integrable into existing methods.

2. Related Work

Test-Time Adaptation (TTA) helps machine learning models perform better on data they have not seen before, without needing labels. When adapting the model, a popular method is to only update certain parameters, especially those in normalization layers. One of the first online TTA methods is called TENT [21], which is designed to minimize the entropy loss of new data. An extension of that, called EATA [14], adds sample filtering and weight regularisation to improve results. DeYO [10] further improves results by introducing a new measure called Pseudo-Label Probability Difference (PLPD). Other methods like TRIBE [20] and Label Shift Adapter [16] address issues with class imbalance, especially when some classes are very rare. SAR [15] tackles multiple issues simultaneously, including mixed data from different domains, small batch sizes, and online class imbalance, slightly relating to the partial class imbalance discussed in this paper.

Class Imbalance is a major issue in deep learning, especially when some classes are much more common than others in the training or test data. Traditional methods like SMOTE [4] balance the data by adjusting the number of samples in majority and minority classes. Algorithms like Focal Loss [11] give more attention to hard examples, helping the model learn rare classes better. Balanced Meta-Softmax [19] improves the traditional softmax function to handle differences between training and testing data more fairly. Another method uses Online Gradient Descent to adapt to shifts in class distribution during training [24]. A simpler yet effective technique known as Logit Adjustment [13] adds a term based on the target class distribution to the model's outputs, inspiring our proposed approach.

3. Motivating Example: TTA Under Partial Class Imbalance

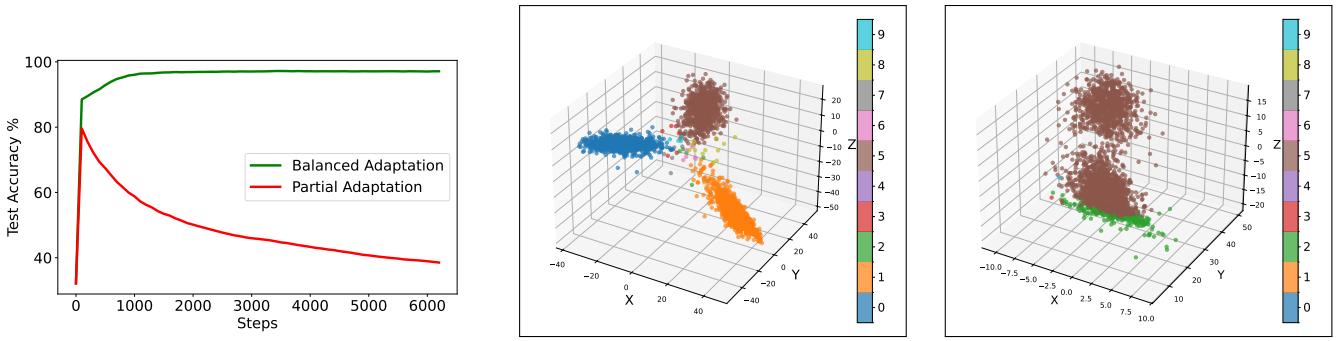
This work is motivated by a significant limitation we have observed in prevalent TTA methods: the performance of entropy minimization deteriorates when adapting to extended test sequences with partial class imbalance. We demonstrate this phenomenon using a controlled experiment with a toy dataset derived from MNIST [9]. Similar patterns arise in larger datasets, but for brevity, we show only one example here. More realistic experiments are provided in Sec. 5. In this example, we first trained the CNN (small two-layer convolutional neural network) on the clean MNIST dataset and then adapted it to a modified target domain, where we added Gaussian noise (severity level 5) to images in the original MNIST validation set[1]. We analysed the model's performance and feature representations when adapting under two scenarios: balanced TTA (all classes present in the unlabeled target domain) and partial TTA (extreme case where samples of only one class are present in the target domain).

The results show that applying TTA under a partial class imbalance setting leads to a performance trajectory where accuracy initially improves but subsequently declines over time (Fig. 1a). This contrasts with balanced TTA, which demonstrates stable performance throughout the adaptation process. Examining the penultimate layer features, we observe that the features of the non-adapted model exhibit significant overlap, leading to poorly defined class boundaries. Applying TTA on a balanced dataset, the features become well separated, clearly distinguishing between classes (Fig. 1b). However, under partial TTA, features from unseen classes tend to shift toward the single visible class, resulting in biased predictions that favour that specific class (Fig. 1c).

3.1. Theoretical Analysis: β Drift under Partial Class Imbalance

In this section, we show how entropy minimization based TTA under *partial class imbalance* induces the batch-normalization shift parameter β to align with the classifier weight of the over-represented class, thereby biasing all predictions and precipitating accuracy collapse.

[1]The noise generation followed the approach described in [8]



(a) (b) (c)

Figure 1. (a) Test accuracy for balanced vs partial TTA, feature space visualization: (b) balanced TTA with all test classes, (c) partial TTA using only digit 5. Features from digits 0, 1, and 5 are shown for clarity.

Consider an arbitrary batch-normalization (BN) layer, h^ℓ. If the activity entering this layer is x^ℓ, then the downstream network can be decomposed as:

$$z = f_\theta(x) = W\psi(h^\ell) + b$$
$$h^\ell = \gamma^\ell \odot \tilde{x}^\ell + \beta^\ell, \qquad \tilde{x}^\ell = \frac{x^\ell - E[x^\ell]}{\sqrt{\text{Var}[x^\ell] + \epsilon}}.$$

Here, the downstream mapping $\psi : \mathbb{R}^{D_\ell} \to \mathbb{R}^D$ subsumes all layers between the BN output h^ℓ and the linear classifier (W, b), and only the affine BN parameters $(\gamma^\ell, \beta^\ell)$ are adapted at test time via entropy minimization using the BN statistics computed on the current test batch. The entropy loss function, $\mathcal{L}$, is defined as:

$$\mathcal{L}(\cdot) = -\sum_{i=1}^{C} p_i \log p_i, \quad p = \text{softmax}(z), \tag{1}$$

where p_i represents the predicted probability for class i.

3.1.1. Gradient with respect to β^ℓ

By the chain rule,

$$\frac{\partial \mathcal{L}}{\partial \beta^\ell} = \left(\frac{\partial h^\ell}{\partial \beta^\ell}\right)^\top \frac{\partial \mathcal{L}}{\partial h^\ell} = \frac{\partial \mathcal{L}}{\partial h^\ell}, \quad \frac{\partial h^\ell}{\partial \beta^\ell} = I$$

$$\frac{\partial \mathcal{L}}{\partial \beta^\ell} = \frac{\partial \mathcal{L}}{\partial h^\ell} = \left(\frac{\partial u}{\partial h^\ell}\right)^\top \frac{\partial \mathcal{L}}{\partial u} = J_\psi(h^\ell)^\top \frac{\partial \mathcal{L}}{\partial u}, \tag{2}$$

where $u := \psi(h^\ell)$ is the classifier input, and $J_\psi(h^\ell) \in \mathbb{R}^{D \times D_\ell}$ is the Jacobian of ψ at h^ℓ.

Lemma 1 (Gradient of entropy loss)**.** *Let $\mathcal{L}(z)$, in Eq. (1), be the entropy loss applied to softmax probabilities. Then the gradient of loss w.r.t. the classifier input u is*

$$\frac{\partial \mathcal{L}}{\partial u} = \sum_{j \in \mathcal{B}} \sum_{i=1}^{C} p_i^j \left(\mathbb{E}_p[z^j] - z_i^j\right) \omega_{i:},$$

where $z = Wu + b$, $\mathcal{B}$ is a batch of sample data points and, $\omega_{i:}$ denotes the i-th row of W.

3.1.2. TTA under Imbalance.

Extreme Imbalance Under extreme class imbalance, where class c dominates ($p_c \approx 1$), this reduces to

$$\frac{\partial \mathcal{L}}{\partial u} \approx \upsilon \omega_{c:}, \quad \upsilon := \sum_{j \in \mathcal{B}} p_c^j (\mathbb{E}_p[z^j] - z_c^j). \tag{3}$$

In practice, during entropy minimization the dominant-class logit z_c often leads the expected logit $\mathbb{E}_p[z]$, so that $\upsilon := \mathbb{E}_p[z] - z_c < 0$. Pushing this gradient back through ψ yields

$$\frac{\partial \mathcal{L}}{\partial \beta_\ell} = J_\psi(h_\ell)^\top \frac{\partial \mathcal{L}}{\partial u} \approx \upsilon \underbrace{J_\psi(h_\ell)^\top \omega_{c:}}_{:= v_c(h_\ell)}.$$

The gradient update with step size $\eta > 0$ is given by $\beta_\ell^{(t+1)} \leftarrow \beta_\ell^{(t)} - \eta \upsilon^{(t)} v_c(h^\ell)$ which indicates that the BN shift parameter β^ℓ is updated in the direction $v_c(h^\ell)$. Assuming a locally frozen Jacobian J_ψ[2] and hence a constant $\upsilon = \upsilon^{(t)} \forall t$ (after the network has converged to the dominant class), it follows that β^ℓ aligns with the dominant pullback vector v_c (Lemma 2).

Lemma 2. *Let $\beta_\ell^{(t)}$ be the batch-normalization shift parameter at iteration t, and let the update rule be given by*

$$\beta_\ell^{(t+1)} = \beta_\ell^{(t)} - \eta \upsilon^{(t)} v_c,$$

where η is nonzero scalar constants, $\upsilon = \upsilon^{(t)} \forall t$, and v_c is a nonzero constant vector. If $\upsilon < 0$, then as the number of iterations T approaches infinity, the vector $\beta_\ell^{(T)}$ aligns with the direction of v_c. That is,

$$\lim_{T \to \infty} \frac{\beta_\ell^{(T)} \cdot v_c}{\|\beta_\ell^{(T)}\| \|v_c\|} = 1.$$

[2]We adopt a local linearisation $J_\psi(h_\ell^{(t)}) \approx J_\psi(h_\ell^{(0)})$ so that the pullback direction $v_c = J_\psi^\top \omega_{c:}$ remains fixed during the drift.

In practice, J_ψ is not strictly constant. If we assume a frozen Jacobian, the analysis shows exact alignment with v_c. More generally, when J_ψ varies smoothly, the same update implies that β^ℓ tracks the evolving target $v_c^{(t)}$, which leads to an approximate alignment.

Moderate Imbalance More typically, test batches are dominated by a small subset of classes $\mathcal{C}^+ \subseteq \{1, \dots, C\}$. For each $i \in \mathcal{C}^+$, define the downstream pullback

$$v_i(h^\ell) := J_\psi(h^\ell)^\top \omega_{i:} \in \mathbb{R}^{D_\ell}.$$

From Eq. (3), the per-class gradient contribution is proportional to $p_i(\mathbb{E}_p[z] - z_i)$, which is typically negative for over-represented classes. The overall update to β^ℓ is therefore

$$\Delta\beta^\ell = -\eta \sum_{i \in \mathcal{C}^+} p_i(\mathbb{E}_p[z] - z_i) v_i(h^\ell).$$

Thus, the effective update direction is a linear combination of the pullback vectors $\{v_i(h^\ell) : i \in \mathcal{C}^+\}$, with coefficients given directly by the entropy-gradient terms $p_i(\mathbb{E}_p[z] - z_i)$. Under the frozen-Jacobian assumption, repeated updates cause β^ℓ to align with this combination:

$$\lim_{T\to\infty} \frac{\beta_\ell^{(T)} \cdot s}{\|\beta_\ell^{(T)}\| \|s\|} = 1, \quad s := \sum_{i \in \mathcal{C}^+} p_i(\mathbb{E}_p[z] - z_i) v_i.$$

In other words, the BN shift parameter aligns with the entropy-weighted sum of the pullback vectors of the over-represented classes.

3.1.3. Effect of β alignment on logits

As shown in Lemma 2, under partial class imbalance entropy minimization aligns β_ℓ with the dominant class pullback vector $v_c := J_\psi(h^\ell)^\top \omega_{c:}$, inducing logit bias.

Lemma 3 (Logit bias under β-alignment)**.** *Assume a frozen downstream Jacobian J_ψ and let the BN shift update follow $\beta_\ell^{(t+1)} = \beta_\ell^{(t)} - \eta\, \upsilon\, v_c$ with step size $\eta > 0$ and scalar $\upsilon < 0$ from the entropy gradient. Then the logits evolve as*

$$z^{(t+1)} \;\approx\; z^{(t)} \;-\; \eta\, \upsilon\, W\, J_\psi J_\psi^\top \omega_{c:}.$$

In particular,

$$\Delta z_c^{(t)} \;=\; -\eta\, \upsilon\, \|J_\psi^\top \omega_{c:}\|_2^2 \;>\; 0,$$

$$\Delta z_k^{(t)} \;=\; -\eta\, \upsilon\, \langle J_\psi^\top \omega_{k:},\, J_\psi^\top \omega_{c:} \rangle \quad (k \neq c).$$

After T steps, the cumulative changes satisfy

$$\frac{z_k^{(T)} - z_k^{(0)}}{z_c^{(T)} - z_c^{(0)}} = \frac{\langle J_\psi^\top \omega_{k:}, J_\psi^\top \omega_{c:} \rangle}{\|J_\psi^\top \omega_{c:}\|^2} \leq \frac{\|J_\psi^\top \omega_{k:}\|}{\|J_\psi^\top \omega_{c:}\|},$$

by Cauchy–Schwarz. Hence the cumulative effect on the dominant logit strictly dominates that of any other class unless $J_\psi^\top \omega_{k:}$ is perfectly aligned with $J_\psi^\top \omega_{c:}$. In particular,

$$\sum_{t=1}^{T} \Delta z_c^{(t)} > \sum_{t=1}^{T} \Delta z_k^{(t)} \quad \forall k \neq c,$$

so entropy minimization systematically amplifies the dominant class logit for any test example more strongly than all others, precipitating collapse under class imbalance.

Proofs for Lemma 1, 2 and 3 are in the supplementary material under the section *Theoretical Analysis*.

4. Method

4.1. Mitigating β-Drift via Logit Adjustment

To prevent β from drifting toward the weight vectors of dominant classes under any degree of class imbalance, we apply a principled logit adjustment [13] before applying the softmax function. For each class i, we redefine its logit as:

$$z_i' = z_i + \tau \log \pi_i \tag{4}$$

where π_i is the target-domain class distribution prior (estimated using the procedure in Sec. 4.2), and $\tau > 0$ is a tunable temperature.

By the same backpropagation through the affine BN step (cf. Sec. 3.1), one obtains the gradient in the expectation-minus-logit form:

$$\frac{\partial \mathcal{L}'}{\partial \beta} = \sum_{i=1}^{C} p_i' \big(\mathbb{E}_{p'}[z'] - z_i'\big)\, \omega_{i:}, \quad \mathbb{E}_{p'}[z'] = \sum_j p_j' z_j'. \tag{5}$$

Decomposing $z_i' = z_i + \tau \log \pi_i$ yields:

$$\mathbb{E}_{p'}[z'] - z_i' = \underbrace{(\mathbb{E}_{p'}[z'] - z_i)}_{\text{(I): drift}} - \underbrace{\tau\big(\log \pi_i - \mathbb{E}_{p'}[\log \pi]\big)}_{\text{(II): correction}}.$$

Substitution into (5) gives:

$$\frac{\partial \mathcal{L}'}{\partial \beta} = \sum_i p_i' (\mathbb{E}_{p'}[z'] - z_i)\, \omega_{i:} - \tau \sum_i p_i' (\log \pi_i - \mathbb{E}_{p'}[\log \pi])\, \omega_{i:}. \tag{6}$$

The above equation allows us to conclude the following:

Extreme Imbalance ($|\mathcal{C}^+| = 1$): Early in adaptation, when π is not yet fully skewed, the adjustment keeps p' less concentrated on c, reducing the magnitude of the drift term by distributing p_i' more evenly. As $\pi_c \to 1$, the drift term approaches $p_c'(\mathbb{E}_{p'}[z] - z_c)\omega_{c:} \approx 0$ (as $\mathbb{E}_{p'}[z] \approx z_c$), but the correction term $-\tau p_c'(\log \pi_c - \mathbb{E}_{p'}[\log \pi])\omega_{c:} \approx 0$ similarly vanishes. This stabilizes β without alignment to $\omega_{c:}$, preserving the ability to correctly classify rare appearances of other classes.

Moderate Imbalance ($|\mathcal{C}^+| > 1$) : For a dominant class i, $z_i > \mathbb{E}_{p'}[z]$ makes the drift term (I) negative along $\omega_{i:}$, while $\log \pi_i - \mathbb{E}_{p'}[\log \pi] > 0$ makes the prior-correction term (II) positive along $\omega_{i:}$. Under gradient descent, (II) opposes (I), reducing the net push toward dominant classes; for under-represented classes the signs flip, increasing their contribution. Thus logit adjustment attenuates β-drift and preserves multi-class separation.

Tuning τ balances drift and correction, while logit adjustment preserves class distinctions and robustness under imbalance.

4.2. Test Prior Calculation

To perform logit adjustment following Eq. (4), a target-domain class prior $\pi \in \Delta^{C-1}$ is required, where Δ^{C-1} denotes the probability simplex over C classes. To this end, we estimate π on-the-fly during test time, updating it with each incoming test batch as detailed below.

Given a mini-batch sample $\{x_j\}_{j=1}^B$ with logits z_{jk} for classes $k \in \{1, \dots, C\}$, the predicted class index for sample x_j is computed as $\hat{y}_j = \arg\max_k z_{jk}$.

Let $n_i = \sum_{j=1}^B \mathbf{1}\{\hat{y}_j = i\}$ be the per-class counts, the batch-wise class prior is computed as:

$$\pi_i^{\text{batch}} = \frac{\sqrt{n_i}}{\sum_{k=1}^C \sqrt{n_k}}.$$

Finally, the running test prior is updated with the current batch prior using the exponential moving average (EMA):

$$\pi^{(t)} = \alpha\, \pi^{(t-1)} + (1-\alpha)\, \pi^{\text{batch}}, \tag{7}$$

where $\alpha \in [0, 1)$ is a hyperparameter to control the prior convergence rate. This procedure allows the class prior to dynamically mimic the evolving target-domain class distribution throughout adaptation.

We initialize $\pi^{(0)}$ to the uniform distribution and renormalize after each update to ensure $\pi^{(t)} \in \Delta^{C-1}$. The resulting $\pi^{(t)}$ is used for logit adjustment in the current batch. In Sec. 6.3, we compare our estimated class prior to the oracle prior, which we calculated using the ground-truth labels.

5. Experimental Details

Datasets: We evaluate our method on three widely adopted benchmark datasets for test-time adaptation, namely CIFAR-100-C, ImageNet-C [8] and DomainNet-126 [17]. CIFAR-100-C and ImageNet-C contain images corrupted with 15 different perturbation types, each applied at 5 severity levels. For each corruption and severity combination, CIFAR-100-C includes 10,000 images corresponding to 100 classes, and ImageNet-C comprises 50,000 images in 1,000 classes. Meanwhile, following the common protocol in test-time adaptation, four different domains from the DomainNet-126 dataset, namely Real(R), Sketch(S), Painting(P), and Clip-art(C) were used in our experiment. Specifically, adaptation from the real domain to the other three was evaluated.

Construction of Partial Class Imbalance Datasets: As outlined in Section 1, we define partial class imbalance as a scenario in which only a subset of the original object classes is present during adaptation. For instance, when adapting a model to an urban environment, only object classes relevant to that setting may be observed. To simulate partial class imbalance using the datasets mentioned in the previous section, we include a subset of classes from the original set, with the size of subset chosen according to the severity of imbalance, which will be detailed in the next section.

A challenge arises because, as the number of sampled classes decreases, the total number of available samples decreases. Consequently, the length of the adaptation process also decreases. To address this, we oversample instances from the chosen classes, extending the length of the partially imbalanced dataset to three times that of the original dataset (e.g., 150,000 images for ImageNet-C), mimicking a long-sequence test-time adaptation scenario (a similar approach to SAR [15]). Additionally, to maintain a balanced distribution among the selected classes, we employ uniform sampling across them when constructing the dataset.

Severity of Partial Class Imbalance Datasets: We control the severity of class imbalance by varying the proportion of object classes observed during test-time adaptation. We define the class proportion as the *Class Inclusion Ratio* (CIR). In our experiments, CIR values of 0.1 and 0.5 correspond to 10% and 50% of the full set of classes, respectively (e.g. 100 and 500 classes for ImageNet-C). Additionally, to examine the extreme limits of test-time adaptation methods, we generate a dataset containing only a single class, and the results are discussed in Sec. 3.1.2.

Implementation Details: On CIFAR-100-C, we use a pretrained ResNeXt29_32×4d from RobustBench [6, 25]. For ImageNet-C, we evaluate a Torchvision ResNet-50 [12] and a ViT-B/16 [7] with Timm [23] pretrained weights. On DomainNet-126, we employ a ResNet-50 trained on the "Real" domain from AdaContrast [5]. To ensure fairness, we follow the learning rates and batch sizes of the original TTA implementations. For ViTs, we apply a 100-step prior warm-up to reduce noisy priors which prevents model collapse in early stages due to incorrect logit adjustment.

Baseline TTA Algorithms: As this work represents the first investigation of test-time adaptation under partial class imbalance, no established benchmark currently exists for evaluating methods in this setting. To address this gap, we introduce a comprehensive benchmark that implements and assesses four state-of-the-art TTA algorithms, namely TENT [21], EATA [14], SAR [15], and DeYO [10]. Our proposed approach is applied in conjunction with each base-

Method	Noise			Blur				Weather				Digital				Avg
	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Brightness	Contrast	Elastic	Pixel	JPEG	
AllAcc (%) ↑																
Unadapted	2.21	2.93	1.85	17.9	9.82	14.8	22.5	16.9	23.3	24.4	58.9	5.43	17.0	20.6	31.7	18.0
COTTA	8.72	8.68	9.81	8.16	10.7	15.1	18.7	18.1	16.6	27.3	39.8	8.62	22.8	30.9	23.5	17.8
DEYO	11.6	10.9	13.6	10.6	11.4	21.4	25.2	26.7	18.8	36.4	46.2	10.1	27.5	34.3	29.3	22.3
TENT	10.8	12.6	11.7	12.4	12.2	22.1	26.3	28.1	21.4	39.4	53.8	4.90	30.3	38.1	31.1	23.7
EATA	12.4	13.3	12.9	11.0	12.0	23.1	28.3	30.2	27.0	42.4	56.7	10.7	33.5	40.4	33.9	25.9
SAR	18.5	20.9	22.7	17.4	18.9	29.5	37.2	38.0	33.4	49.3	60.0	14.9	39.4	47.9	41.3	32.6
EATA + LA	17.5	19.1	18.5	17.7	17.2	18.8	37.6	25.3	29.0	47.6	58.2	6.86	41.2	46.7	39.5	29.4
COTTA + LA	17.1	16.8	17.8	13.5	16.9	23.7	33.1	32.1	29.8	43.9	59.0	15.7	40.9	46.7	37.9	29.7
DEYO + LA	16.4	19.9	17.6	16.8	17.1	28.1	34.5	33.2	29.0	45.1	57.3	15.6	37.7	45.7	39.8	30.3
TENT + LA	19.0	22.1	20.5	20.1	20.1	30.6	41.1	38.3	34.4	52.1	64.4	13.7	43.9	52.0	43.7	34.4
SAR + LA	20.3	22.6	21.6	20.3	20.2	32.5	39.3	39.7	33.1	51.5	63.7	14.2	42.7	51.1	44.0	34.5
PeAR (%) ↓																
TENT	69.6	60.2	64.9	54.7	55.4	39.1	54.1	38.6	65.4	30.4	21.7	289.8	46.9	34.3	38.7	64.2
COTTA	59.7	58.2	44.1	36.8	36.4	42.1	54.2	62.0	62.5	55.0	39.9	37.8	52.1	39.2	46.8	48.5
DEYO	63.6	73.4	45.2	51.0	48.7	32.0	40.7	35.1	68.7	27.8	25.2	40.6	36.1	34.1	32.6	43.6
EATA	15.3	8.21	11.0	24.5	15.3	15.8	32.8	21.5	23.6	18.3	13.4	68.8	27.6	24.8	20.0	22.7
SAR	5.84	7.93	6.59	6.63	6.50	5.06	11.7	6.66	8.93	7.66	9.04	25.8	13.5	8.51	7.40	9.19
EATA + LA	6.36	13.7	9.35	12.6	9.85	31.9	11.7	26.0	15.1	5.69	5.93	81.5	7.29	7.71	9.68	17.0
DEYO + LA	22.3	20.0	20.6	18.1	19.6	14.1	14.7	16.3	20.3	9.72	10.0	24.4	11.3	11.7	8.9	16.1
SAR + LA	2.74	2.23	2.32	3.60	1.83	2.18	4.30	2.33	8.61	2.43	2.93	37.0	4.83	2.28	1.74	5.43
TENT + LA	2.81	2.21	2.66	3.49	3.28	4.16	1.70	2.92	4.00	0.87	1.70	34.6	3.00	1.25	1.49	4.68
COTTA + LA	5.02	4.37	6.22	5.36	5.31	4.16	3.20	3.45	3.10	2.84	1.44	10.19	2.50	2.02	2.31	4.10

Table 1. Partial class imbalance benchmarking at Class Inclusion Ratio (CIR) set to 0.1 (100 classes) on the ImageNet-C dataset with corruption level 5. We compare baseline TTA methods and their counterparts enhanced with our logit adjustment (LA) method, reporting both All-Class Accuracy (AllAcc) and Peak-to-Average Reduction (PeAR) across all 15 corruption types.

line algorithm, and performance improvements are measured relative to the respective original methods. These results demonstrate the efficacy of our approach in enhancing the robustness of existing test-time adaptation algorithms when confronted with partial class imbalance. Additionally, we include a non-adaptive model, which does not incorporate any test-time adaptation, as a reference for comparison.

Evaluation Metrics: To assess performance under partial class imbalance, we introduce two evaluation metrics designed to capture both the accuracy and stability of test-time adaptation algorithms. These metrics are *All-Class Accuracy (AllAcc)* and *Peak to Average Reduction (PeAR)*, which will be detailed next.

The All-Class Accuracy (AllAcc) metric provides a comprehensive measure of a model's performance across the entire set of object classes. For each dataset under consideration, we construct a hold-out test set that includes all classes (both those observed and unobserved during partial test-time adaptation). For the observed classes, the test set contains only samples that are not used for adaptation, thus ensuring an unbiased evaluation of generalization. This setup enables us to capture not only a model's ability to maintain accuracy on the classes it has seen, but also its capacity to retain knowledge of unobserved classes, revealing the occurrences of partial forgetting. Such forgetting occurs when adaptation leads to strong performance on observed classes, but severely degraded accuracy for unobserved ones, which is highly undesirable in practice.

While AllAcc reports average accuracy over time, it cannot distinguish between methods that quickly reach optimal performance and then degrade, and those that maintain stable accuracy throughout adaptation. This distinction is critical, as persistent degradation after reaching peak performance may ultimately result in model collapse. To this end, we introduce the Peak to Average Reduction (PeAR) metric that quantifies the stability of the adaptation process itself, which was motivated by the crest factor [3] commonly used in signal processing. Specifically, PeAR is defined as:

$$\text{PeAR} = \left(\frac{\text{Peak Test Accuracy}}{\text{AllAcc}} - 1 \right) \times 100\% \quad (8)$$

where Peak Test Accuracy is the maximum accuracy captured along the test-time adaptation process. A higher PeAR value indicates greater instability, indicating that a method's performance deteriorates after its initial peak. Combining AllAcc and PeAR results in a comprehensive evaluation framework that captures both the effectiveness and reliability of TTA algorithms under partial class imbalance.

6. Experimental Results and Discussion

6.1. State-of-the-art Benchmark for Partial Class Imbalance

This section compares the performance of the baseline test-time adaptation (TTA) algorithms and their respective variants improved with our proposed logit adjustment mechanism. Evaluations are conducted using our custom

Method	TENT		EATA		DEYO		SAR	
LA	✗	✓	✗	✓	✗	✓	✗	✓
CIFAR-100-C (Corruption Level 3; CIR=0.01)								
AllAcc	62.1	**64.0**	68.6	**70.2**	58.1	**61.6**	**69.4**	**69.4**
PeAR	12.5	**9.17**	2.36	**0.03**	20.2	**13.4**	**1.35**	1.43
CIFAR-100-C (Corruption Level 3; CIR=0.1)								
AllAcc	62.3	**65.1**	65.9	**66.8**	59.9	**64.1**	69.6	**70.5**
PeAR	14.4	**9.32**	6.59	**5.50**	18.4	**11.3**	1.60	**0.88**
CIFAR-100-C (Corruption Level 5; CIR=0.1)								
AllAcc	56.7	**59.6**	60.3	**60.9**	54.6	**58.4**	64.3	**65.1**
PeAR	16.5	**10.8**	7.31	**6.40**	20.2	**13.1**	1.91	**1.12**
ImageNet-C (Corruption Level 3; CIR=0.1)								
AllAcc	23.7	**34.4**	25.9	**29.4**	22.3	**30.3**	32.6	**34.5**
PeAR	64.2	**4.68**	22.7	**17.0**	43.6	**16.1**	9.19	**5.43**
DomainNet-126 (CIR=0.1)								
AllAcc	54.5	**55.0**	51.7	**52.3**	52.2	**54.1**	55.6	**56.4**
PeAR	2.80	**1.79**	4.74	**3.71**	7.60	**4.57**	**0.43**	0.80

Table 2. Comparison of baseline methods and their counterparts enhanced with our logit adjustment (LA) method, under partial class imbalance with different levels of CIR. Results are reported on CIFAR-100-C and ImageNet-C with corruption levels 3 and 5, as well as on the DomainNet-126 dataset. For the DomainNet-126 dataset, domain shifts occur from Real(R) to Sketch(S), Painting(P) and Clipart(C).

datasets created to simulate partial class imbalance. Specifically, experiments are performed under the most severe imbalance condition (CIR=0.1) across three benchmarks, namely CIFAR-100-C, ImageNet-C, and DomainNet-126. For CIFAR-100-C and ImageNet-C, results covering 15 corruption types at two severity levels (3 and 5) are reported. For DomainNet-126, we report results across three domain shifts using real-world images as the source domain.

The results for ImageNet-C, summarized in Tab. 1, highlight the effectiveness of our proposed logit adjustment mechanism under severe partial class imbalance. While TENT yields an improved AllAcc compared to the unadapted source model, its adaptation process is highly unstable, as indicated by a Peak-to-Average Reduction (PeAR) of 64.2%. In contrast, by integrating our proposed logit adjustment method with the TENT algorithm (TENT + LA), we have achieved a substantial performance boost, attaining an AllAcc of 34.4% while PeAR drops to 4.68%, reflecting greater stability. As a complementary study, we incorporate logit adjustment into the teacher–student TTA method COTTA [22], and show that our approach substantially enhances models with very low baseline performance.

Moreover, we show that even strong-performing TTA methods, such as SAR [15], benefit from the integration of our approach. Incorporating our logit adjustment into SAR improves its AllAcc from 32.6% to 34.5% while reducing PeAR from 9.19% to 5.43%. In addition, a similar improvement trend is observed with the Vision Transformer (ViT-B) model, as reported in Tab. 3.

Beyond ImageNet-C, our method also achieves impressive and consistent improvements across all included baseline TTA algorithms, on CIFAR-100-C and DomainNet-126. As shown in Tab. 2, the integration of our logit adjustment mechanism leads to notable gains in both All-Class Accuracy (AllAcc) and adaptation stability (as measured by PeAR) for every evaluated baseline. These results collectively demonstrate that our method is both effective and method-agnostic, as it can be seamlessly integrated into diverse test-time adaptation algorithms across different datasets, while consistently enhancing both adaptation accuracy and stability.

Method	TENT		EATA		DEYO		SAR	
LA	✗	✓	✗	✓	✗	✓	✗	✓
AllAcc	57.1	**59.7**	56.7	**59.9**	60.9	**63.8**	60.7	**64.0**
PeAR	95.2	**1.34**	139.0	**0.21**	**2.07**	3.12	4.83	**2.05**

Table 3. Comparison of baseline methods and their counterparts enhanced with our logit adjustment (LA) method under partial class imbalance at CIR = 0.1. Results are obtained using the Vision Transformer (ViT-B) model on the ImageNet-C dataset with corruption level 5.

6.2. Adaptation under Extreme Class Imbalance

Beyond standard partial class imbalance evaluations, we also assess our approach in an extreme scenario where only a single class is observed (CIR = 0.01) during test-time adaptation on CIFAR-100-C. As shown in Tab. 2, integrating our logit adjustment mechanism consistently enhances both accuracy and adaptation stability across all baseline TTA methods and corruption severities. For instance, with EATA, our method improves AllAcc from 68.6% to 70.2% while reducing PeAR from 2.36% to 0.03%; for DEYO, it improves AllAcc from 58.1% to 61.6% and lowers PeAR from 20.2% to 13.4%. These results demonstrate the robustness and generality of our approach, showing strong performance even under the most adverse class imbalance conditions.

6.3. Discussion

Impact of Partial Class Imbalance Severity on TTA Performance: To further analyze the influence of partial class imbalance on test-time adaptation, we evaluate how varying the severity of class imbalance affects the overall performance. Specifically, the severity of class imbalance is measured as the proportion of observable classes, as discussed in Sec. 5. In Fig. 2, we show the changes in AllAcc and PeAR, for a range of TTA algorithm baselines enhanced with our logit adjustment mechanism, compared to the unadapted source model. Notably, as the severity level continues to increase (fewer classes are observed during adaptation), our method improves the baselines and attains stable performance, even in the most adverse class imbalance scenarios. These results confirm that, while the severity of partial class imbalance may impact adaptation success, our

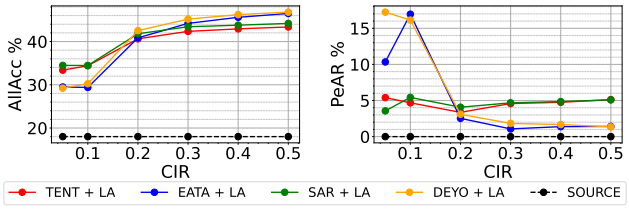


Figure 2. AllAcc % (left) and PeAR % (right) with varying CIRs.

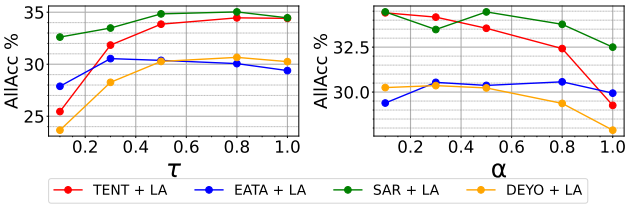


Figure 3. AllAcc % variation with τ (left) and α (right).

method delivers robust and reliable accuracy improvements irrespective of the underlying adaptation baseline, providing clear benefits in challenging settings.

Hyperparameter Sensitivity Analysis The proposed logit-adjusted entropy loss, which was defined in Eq. (4), involves two hyperparameters, namely τ and α. The hyperparameter τ modulates the strength of the logit adjustment. Thus, lower values of τ reduce the influence of logit adjustment, leading to a decrease in overall TTA performance, as illustrated in Fig. 3.

The hyperparameter α, as introduced in Eq. (7), governs how the target-domain class distribution prior $\pi^{(t)}$ is updated using an exponential moving average (EMA). Lower values of α adapt quickly to the current batch, enabling rapid convergence but with higher variance under class imbalance. In contrast, higher values of α update conservatively, emphasizing historical information over recent data, which reduces noise sensitivity but slows adaptation to distribution shifts. As evidenced in Fig. 3, a consistent trend emerges, where larger values of α generally lead to deterioration in model accuracy across all methods. This result underscores the necessity of appropriate hyperparameter tuning, as overly conservative priors can impair adaptation by failing to adequately respond to new test-time distributions, while too aggressive updates may amplify noise.

In all of our experiments (unless stated otherwise), we set $\tau = 1.0$ and $\alpha = 0.1$, which provides a balanced trade-off between TTA accuracy and stability.

Evaluation of Estimated Class Prior π In this section, we evaluate the quality of our estimated target-domain class prior π with the oracle prior π^*, which we calculated using the ground-truth class labels $\mathcal{Y}^t$. Specifically, π^* of class i is computed as

$$\pi_i^* = \begin{cases} 1/|\mathcal{C}^+|, & \text{if } i \in \mathcal{Y}^t, \\ \varepsilon, & \text{otherwise} \end{cases}$$

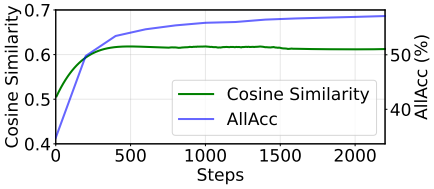


Figure 4. Cosine similarity between the estimated and oracle priors, during adaptation. Results are obtained using the ViT-B model on ImageNet-C dataset with level 5 glass blur corruption.

where $\mathcal{C}^+$ is the subset of classes in the target domain dataset (e.g. 100 for ImageNet-C with CIR = 0.1) and ε is an extremely small value which we set to 1×10^{-12}.

To quantitatively assess the similarity between the estimated prior and the oracle prior, which informs us about the alignment with the true target-domain class distribution, we compute the cosine similarity at each adaptation step as follows:

$$\text{sim}(\pi^*, \pi^{(t)}) = \frac{\pi^* \cdot \pi^{(t)}}{\|\pi^*\| \, \|\pi^{(t)}\|},$$

where $\cdot$ denotes the dot product and $\|\cdot\|$ denotes the Euclidean norm. As shown in Fig. 4, the cosine similarity between our estimated prior and the oracle prior steadily increases and eventually plateaus as adaptation progresses. Notably, the test accuracy exhibits a similar stabilization trend alongside the cosine similarity, implying that improved alignment of the estimated prior with the true class distribution facilitates more reliable and stable model performance during adaptation.

We further study two strategies, namely using predicted class indices versus softmax outputs, to update the test prior (see Sec. 4.2). Both variants outperform the baseline, while using the predicted class indices yields sharper estimates and faster convergence. Results and detailed discussion are included in the supplementary due to space constraints.

7. Conclusions

In this work, we investigated the vulnerability of Test-Time Adaptation (TTA) methods to partial class imbalance, a scenario where only a subset of classes dominate the test stream. We showed that, under such conditions, TTA initially improves performance but ultimately degrades due to normalization layers driving the model toward collapse. To address this, we introduced a logit-adjusted entropy minimization framework that stabilizes adaptation and mitigates the adverse effects of imbalance. Extensive experiments across datasets, corruption types, architectures, and TTA methods confirm that partial class imbalance presents a fundamental challenge to reliable adaptation, and our approach consistently achieves higher accuracy while ensuring stable adaptation.

References

[1] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018. 1

[2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 1

[3] S. Boyd. Multitone signals with low crest factor. *IEEE Transactions on Circuits and Systems*, 33(10):1018–1022, 1986. 6

[4] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority oversampling technique. *J. Artif. Int. Res.*, 16(1):321–357, 2002. 2

[5] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 295–305, 2022. 5

[6] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 5

[7] 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. 5

[8] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 2, 5

[9] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 2

[10] Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 5

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[12] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision, 2016. 5

[13] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021. 2, 4

[14] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022. 2, 5

[15] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 5, 7

[16] Sunghyun Park, Seunghan Yang, Jaegul Choo, and Sungrack Yun. Label shift adapter for test-time adaptation under covariate and label shifts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16421–16431, 2023. 1, 2

[17] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 5

[18] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. 1

[19] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2020. Curran Associates Inc. 2

[20] Yongyi Su, Xun Xu, and Kui Jia. Towards real-world test-time adaptation: Tri-net self-training with balanced normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 15126–15135, 2024. 1, 2

[21] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 1, 2, 5

[22] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7201–7211, 2022. 1, 7

[23] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 5

[24] Ruihan Wu, Chuan Guo, Yi Su, and Kilian Q. Weinberger. Online adaptation to label distribution shift. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2021. Curran Associates Inc. 2

[25] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 5