Test-Time Adaptation on Noisy Data via Model-Pruning-Based Filtering and Flatness-Aware Entropy Minimization

Xingzhi Zhou[1*], Zhiliang Tian[2†], Boyang Zhang[2], Yibo Zhang[2], Ka Chun Cheung[3], Simon See[3], Hao Yang[2], Yun Zhou[2], Nevin L. Zhang[1]

[1]The Hong Kong University of Science and Technology, Hong Kong SAR, China
[2]National University of Defense Technology, Changsha, China
[3]NVIDIA AI Technology Center, NVIDIA
{xzhoubl, lzhang}@cse.ust.hk {tianzhiliang, zhangboyang, zhangyibo22, yanghao, zhouyun}@nudt.edu.cn
{chcheung,ssee}@nvidia.com

Abstract

Test-time adaptation (TTA) deals with domain shifts during inference by training models based on only unlabeled test samples. Test samples may include noisy samples, which degrade domain adaptation. Existing methods rely on the model's output prediction to detect and filter noisy samples, and further search for flat regions during optimization, which makes the optimization more robust on noisy samples. However, there are two issues: (1) the output prediction tends to be inaccurate due to domain shifts, weakening noisy-sample detection; (2) current approaches for searching flat regions focus on optimization to enhance the worst case, which ignores achieving flatness by avoiding the quick changing of losses. To address these challenges, we propose a **mo**del pruning-based **t**est-**t**ime **a**daptation model for noisy data streams, named MoTTA, which leverages a new proposed filtering, output difference under pruning (ODP)-based filtering, and a **flat**ness-aware **e**ntropy **m**inimization (FlatEM). Specifically, to reduce the impact of inaccurate output predictions, ODP-based filtering measures the output difference of a sample before and after model pruning, which works even under inaccurate output. To improve the search for flat loss surfaces, FlatEM integrates zeroth-order flatness and first-order flatness (minimize the maximal gradient normalization with a weight perturbation constrained in a small Euclidean ball) on entropy minimization. To solve these hard maximum problems, we leverage Taylor expansion to obtain approximated results for optimization. FlatEM also adopts a parameter regularization to mitigate incorrect updates from noisy samples. The experiments show our advantages in dealing with noisy data streams at TTA comparable to existing baselines.

Code — github.com/XingzhiZhou/MoTTA

Introduction

Unsupervised domain adaptation (UDA) addresses domain shifts by using transfer learning to apply knowledge from labeled source domains to unlabeled target domains (Long et al. 2015; Ganin et al. 2016). Domain shifts refer to the variations between the source and target domains. However, accessing labeled source-domain data could be challenging due to computational overhead or privacy concerns, and collecting target-domain data requires significant human efforts. Test-time adaptation (TTA) is a specialized approach within UDA that aims to tackle these challenges.

TTA is required not to access source-domain data and instead adapts the model using only unlabeled test samples, which are drawn from the target domain experiencing domain shifts (Liang, He, and Tan 2024). TTA enhances model performance at inference, making it useful for deploying models in situations where the target domain distribution is unpredictable, e.g., medical image classification across different environments (Kim et al. 2022). The primary strategy of TTA is to continuously learn from test samples, thereby enhancing the generalization ability of the model.

TTA falls into two main categories (Niu et al. 2024): gradient-free and gradient-based methods. (1) Gradient-free methods learn from test samples without conducting back-propagation, instead adjusting the model's modules or outputs. The adjustments include updating batch normalization statistics (Schneider et al. 2020; Lim et al. 2023), the classifier layer (Iwasawa and Matsuo 2021) (modules), or refining the predicted probability (Boudiaf et al. 2022) (outputs) based on test samples. These methods freeze most model parameters, thereby reducing the risk of catastrophic forgetting on the source domain. However, the constraints of freezing the primary model parameters limit the learning capacity. (2) Gradient-based methods update the model parameters in real time through self-supervised or unsupervised learning over test data. The objective functions for these learning tasks include entropy minimization (Wang et al. 2021), teacher-student self-training (Yuan, Xie, and Li 2023), and contrastive learning (Chen et al. 2022). The potential risk with these methods is overfitting to the current samples, which results in catastrophic forgetting on the source domain and losing the generalization ability to adapt to continually arriving new domains. Both gradient-free and gradient-based methods assume that the categories of test samples fall within the source domain. However, in real world, test samples may contain categories out of the source domains, referred to as *noisy samples*, which reduces the effectiveness of these TTA

[*]This work was done while Xingzhi was an intern at NVIDIA.
[†]Corresponding author.

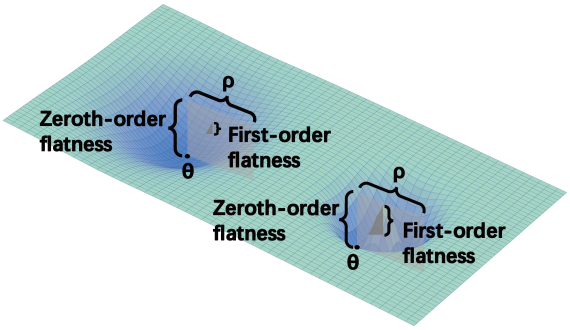


Figure 1: A toy case study on zeroth-order flatness versus first-order flatness. When quick changing of loss exists within a perturbation radius ρ, zeroth-order flatness may not reliably reflect current flatness. In contrast, first-order flatness provides a more accurate measurement.

methods (Gao, Zhang, and Liu 2024).

To handle noisy samples in TTA, researchers use model-prediction-based methods (Lee et al. 2023; Gao, Zhang, and Liu 2024) to filter out noisy samples by considering the output predictions (probabilities or logits) during adaptation. However, domain shifts between source and target domains degrade the model and result in inaccurate output probabilities, which weaken noisy sample filtering and lead to model drift. Further, researchers propose flatness-aware model-prediction methods (Gong et al. 2023) to refine the model-prediction-based methods by searching for flat regions in optimization to make optimization more robust, thereby mitigating the incorrect updates caused by noisy samples. However, flatness-aware model-prediction focuses solely on optimizing to enhance the worst case within a weight perturbation radius (i.e., zeroth-order flatness (Foret et al. 2021)), which is insufficient for adequately measuring flatness in a scenario with quick changing of the loss, shown in Fig. 1.

To address the above issue, we propose MoTTA, a **mo**del pruning-based **t**est-**t**ime **a**daptation model designed for noisy data streams with domain shifts, which incorporates an output difference under model pruning (ODP)-based filtering and a flatness-aware entropy minimization (FlatEM). The ODP-based filtering mitigates the impact of inaccurate prediction in noisy sample filtering, while FlatEM improves the search for flat surfaces to avoid sharp drops in optimization (large slopes on the loss surface, as Fig. 1), enhancing adaptation performance. Specifically, **ODP-based filtering** measures the output difference of samples before and after model pruning, and filters samples with this score. ODP-based filtering can effectively distinguish noisy samples even if the original prediction is inaccurate under domain shifts, since this filtering relies on the output difference instead of solely on the output prediction. **FlatEM** employs an entropy loss that incorporates flatness constraints and parameter regularization. The flatness constraints force the model to account for the worst case (zeroth-order flatness) and the maximum gradient norm (first-order flatness) with a weight perturbation constrained by a small Euclidean ball. First-order flatness captures quick changes in the loss surface by measuring significant maximum gradient norms. To relieve the difficulty in optimizing the flatness constraints, we apply Taylor expansion to get the approximated gradients for optimization. The parameter regularization further helps to prevent the incorrect fitting on noisy samples. We validate our model on three TTA benchmarks (ImageNet-C, ImageNet-R, ImageNet-Sketch) using two datasets (SSB-hard, NINCO) as noisy datasets.

Our contributions are: (1) We propose a model-pruning-based sample filtering to identify noisy samples dealing with inaccurate predictions under domain shifts. (2) We propose flatness-aware entropy minimization (FlatEM), which integrates zeroth-order and first-order flatness to improve the search for flat loss regions. (3) We empirically validate the effectiveness of MoTTA on three target datasets with two noisy datasets to provide noisy samples separately.

Related Works

Test-Time Adaptation Test-time adaptation adapts the model to deal with the domain shifts during inference with online unlabeled data. Test-time adaptation falls into two categories. (1) *Gradient-free methods* update modules or model output with only inference (Boudiaf et al. 2022; Niu et al. 2024; Iwasawa and Matsuo 2021). (2) *Gradient-based methods* utilize the gradient of objective functions to update model parameters based on test samples. TENT (Wang et al. 2021) uses entropy minimization to enhance adaptation. To improve TENT, EATA (Niu et al. 2022) filters out test data to ensure low uncertainty and high diversity, and regularizes the parameters weighted by Fisher importance (Kirkpatrick et al. 2017) to avoid catastrophic forgetting. To deal with continual domain shifts, CoTTA (Wang et al. 2022) adopts a teacher-student training framework with averaged data augmentation, as well as restoring partial parameters to mitigate overfitting. VIDA (Liu et al. 2024) further improves COTTA by considering low-rank and high-rank learnable weights to facilitate adaptation. However, these methods have potential overfitting risk of obtaining overconfident prediction results (Yang et al. 2024a,b). TEA (Yuan et al. 2024) solves this issue by modeling an energy model to reduce calibration errors during adaptation.

Test-Time Adaptation on Noisy Data Stream In real-world scenarios, test-time samples might contain categories outside the source domain, inspiring research for TTA on noisy data streams. A straightforward approach is to leverage the methods for out-of-distribution (OOD) detection to identify the noisy samples, specifically using score-function-based approaches (Bendale and Boult 2016; Hendrycks and Gimpel 2017; Liang, Li, and Srikant 2018; Lakshminarayanan, Pritzel, and Blundell 2017). However, in TTA tasks, we need to distinguish noisy samples from domain-shifted target samples and update the model on the fly, making OOD detection less effective. Pioneering work in TTA on noisy data streams, OSTTA (Lee et al. 2023) identifies noise data by analyzing the predicted confidence difference along the adaptation process. UniEnt (Gao, Zhang, and Liu 2024) measures the cosine similarity of features

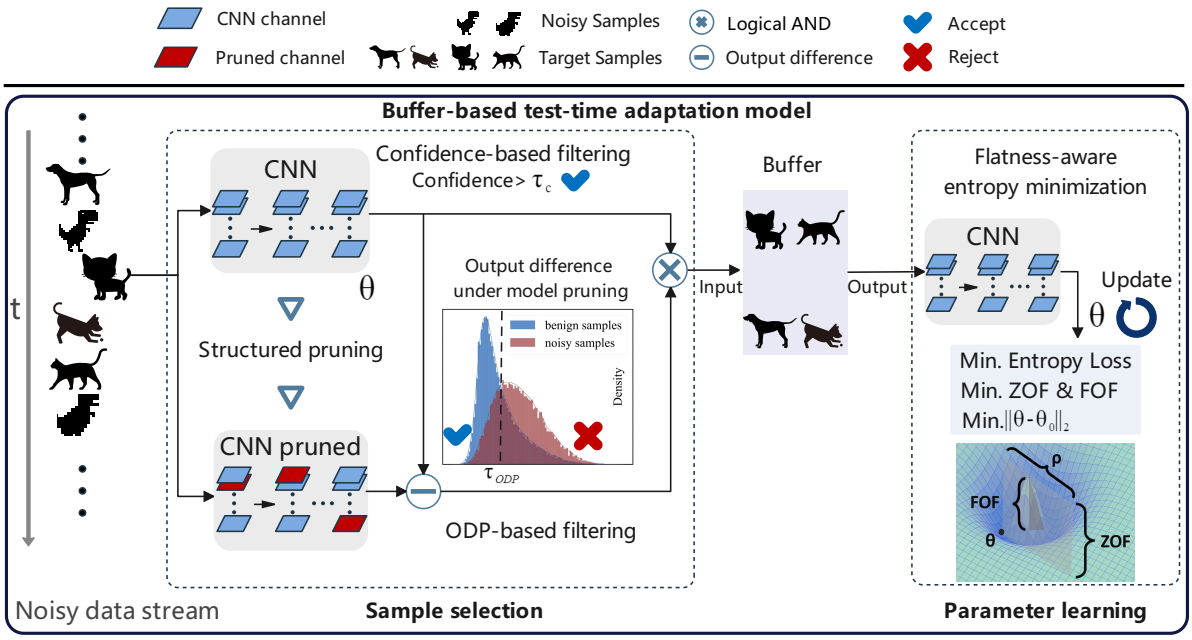


Figure 2: MoTTA framework. For each test sample (like this cat in the figure), we first measure its predicted confidence and output difference under pruning, and then filter samples according to these two scores. Second, we insert the filtered samples into a buffer. Third, we take samples from the buffer to train the model θ on an entropy loss with zeroth-order flatness (ZOF), first-order flatness (FOF), and a parameter regularization.

and weights in the classifier and uses the Gaussian mixture model to filter out noisy samples, adapting the model with an entropy loss. Expanding predicted confidence filtering, SoTTA (Gong et al. 2023) applies sharpness-aware entropy minimization (Foret et al. 2021; Li et al. 2024) to prevent misleading parameter updates caused by noisy samples.

Open-Set Domain Adaptation Open-set domain adaptation (OSDA) aims to transfer knowledge from a labeled source domain to an unlabeled target domain, with the assumption of unknown categories in the target domain. OSDA falls into two categories: (1) *Sample separation-based methods* discriminate unknown categories using predefined metrics (Liu et al. 2019; Wan et al. 2024; Bucci, Loghmani, and Tommasi 2020). (2) *Aligning the cross-domain known distribution* emphasizes on domain alignments through adversarial learning (Saito et al. 2018) and graph learning (Luo et al. 2020). TTA on noisy data streams is a special setting for ODSA at inference. However, we cannot directly use ODSA methods for the noise sample problem in TTA, since ODSA relies on labeled source data and unlabeled target data during the training phase, which is different from the TTA setting.

Method

Problem Definition

Given a classification model f_{θ_0} with initial parameters θ_0 pre-trained on a source domain with the dataset $\{x_i^s, y_i^s\}_{i=1}^{N_s} \sim \mathcal{P}_s$, where $\mathcal{P}_s$ is the source domain distribution, the test-time adaptation on noisy data streams aims to adapt f_{θ_0} to online unlabeled samples $x_1, x_2, ..., x_T$ with target domain distribution $\mathcal{P}_t$ and domain shifts, where partial samples are noisy samples (samples with categories outside source domain $\mathcal{P}_s$).

Overview

MoTTA is based on a buffer-based test-time adaptation model with a sample selection and parameter learning, as shown in Fig. 2.

- **Buffer-based test-time adaptation model** (BTAM) adapts a classification model pre-trained before test-time adaptation (TTA) (referred to as the pre-trained source model) to noisy data streams by selectively storing test samples within a buffer and updating the model using the saved data.
- **Sample selection via output difference under model pruning** filters test samples according to the predicted confidence and the output difference of a sample before and after model pruning.
- **Parameter learning via flatness-aware entropy minimization** updates the model parameters by searching for flat regions on a training loss surface and constrains the model parameters from initial parameters.

BTAM is our backbone. The sample selection filters samples for BTAM. BTAM provides samples from its buffer for the parameter learning on the pre-trained source model.

Buffer-Based Test-time Adaptation Model (BTAM)

To adapt the pre-trained source model to noisy data streams during inference, we employ a buffer-based test-time adap-

tation model (BTAM) that selectively saves test samples in a buffer and uses the saved samples for training to adapt the model at the test time. BTAM belongs to the flatness-aware model-prediction methods (Gong et al. 2023). The adaptation process involves three steps for handling test samples: (1) *Data filtering*: BTAM measures each incoming test sample x according to the confidence of the model predictions and selects samples based on the confidence. This step aims to filter out noisy samples and ensure the quality of selected samples. (2) *Data storing*: BTAM stores the selected sample x into a buffer, along with its predicted label $\hat{y}$. The pre-trained source model infers the predicted label $\hat{y}$ by selecting the class that maximizes the predicted probability $\hat{p}(y|x)$. The buffer management consists of two operations: a) if the buffer is not full when a newly selected sample is filtered, the buffer stores this sample. b) If the buffer is full when a newly selected sample arrives, we use category-balanced sampling from (Gong et al. 2023; Yuan, Xie, and Li 2023) to randomly remove a sample in the predicted class with the largest number of samples and store the new sample. (3) *Model updating*: BTAM updates the parameters of the pre-trained source model by training the samples from the buffer. Through this parameter updating in a data replay style, the pre-trained source model adapts to the target domain in noisy data streams during inference.

Sample Selection via Output Difference under Model Pruning (ODP)

To distinguish noisy samples under domain shifts, we select samples with two filtering strategies as follows.

Output Difference under Model Pruning (ODP)-Based Filtering To effectively filter out noisy samples, we propose output difference under model pruning (ODP)-based filtering that filter samples considering the small output difference before and after conducting model pruning on the pre-trained source model f_θ.

The motivation for ODP-based filtering is to handle the inaccurate model prediction under domain shifts. Existing prediction-based methods use the model prediction directly to identify noisy samples. However, the prediction is inaccurate under domain shifts, degrading the effectiveness of identifying noisy samples. ODP-based filtering distinguishes noisy samples by measuring the output difference before and after pruning low-magnitude weights, rather than relying solely on the output prediction. Even when the model's prediction is incorrect or performs poorly, this approach still enables ODP-based filtering to identify noisy samples by examining the difference. In short, this capability allows ODP-based filtering to reduce the impact of inaccurate predictions under domain shifts.

ODP-based filtering assigns an output difference under model pruning (ODP) score and selects samples with ODP scores lower than a threshold τ_{ODP}. $ODP_\theta(x)$ denotes an ODP score to a sample x as Eq. 1 and the pre-trained source model parameters θ. We use a structured pruning technique for model pruning, which selectively removes out-channels in the convolutional layers of the pre-trained source model f_θ based on their L1 normalization (details in App. G). Let z and z' be the representation of the sample x without the model pruning (i.e. original model) and the representation of sample x with the model pruning on the pre-trained source model, respectively. We define $ODP_\theta(x)$ as the maximum angle difference between with and without model pruning. Each angle difference is the summary of two angles: (1) $\angle(z, w_i)$: the angle between the representation z and the i-th weight w_i in the classifier of the pre-trained source model with total C classes, (2) $\angle(z', w_i)$: the angle between the pruned representation z' and the weight w_i. We use $\angle(a, b) = \arccos \frac{a \cdot b}{||a|| \cdot ||b||}$ to calculate the angle between any vectors a, b. We obtain $ODP_\theta(x)$ as:

$$ODP_\theta(x) = \max_{i \in \{1,2,\ldots,C\}} |\angle(z, w_i) - \angle(z', w_i)| \quad (1)$$

Confidence-Based Filtering To improve sample quality and provide less noisy samples, we use confidence-based filtering to select samples with high predicted confidence by the pre-trained source model f_θ. Specifically, we measure the confidence of predicting a sample x, and select the sample if its confidence is beyond a threshold τ_c. We regard the maximum value of the predicted probability, $\max_y \hat{p}(y|x; \theta)$, as confidence, where θ denotes the pre-trained source model parameters.

Parameter Learning via Flatness-Aware Entropy Minimization

To alleviate the incorrect updates of parameters caused by noisy samples, we apply parameter learning to update the pre-trained source model parameters on test data, involving proposed flatness-aware entropy minimization (FlatEM). FlatEM minimizes an entropy loss on samples with two types of regularization: 1) *flatness constraints* make the model focus on the worst case (zeroth-order flatness) and the maximal gradient normalization (first-order flatness) within a constrained weight perturbation. The existing work (Gong et al. 2023) only considers the zeroth-order flatness, which could not be sufficient to reflect flatness when meeting multiple maximals with large slopes (quick variation on the surface, as Fig. 1). The first-order flatness addresses this issue by focusing on the maximum gradient norm, effectively capturing large slopes through significant gradient values. This inspires us to consider combining first-order flatness to improve generalization ability and relieve the misleading parameter updates caused by noisy samples. 2) *parameters regularization* enforces the pre-trained source parameters close to the original parameters to reduce the impact of noisy samples and relieve potential overfitting.

Specifically, $\mathcal{L}(x, \theta)$ denotes the total loss in flatness-aware entropy minimization on a sample x and the pre-trained source parameters θ as Eq. 4, which contains three modules, an entropy loss, flatness constraints, and a parameter regularization.

$H(y|x; \theta)$ denotes an entropy loss on the predicted probability $\hat{p}(y|x; \theta)$ for sample x with parameters θ.

Flatness constraints contain two orders of flatness to search for flat regions on the entropy loss surface: zeroth-order flatness and first-order flatness as follows (Zhang et al. 2023b).

Zeroth-order flatness $\mathcal{R}_0(x; H, \theta)$ is based on sample x, the entropy loss H, and the pre-trained source model parameters θ as Eq. 2. We first find a weight perturbation ϵ applied to the parameters θ to maximize the entropy loss $H(y|x; \theta + \epsilon)$, where ϵ is constrained within a Euclidean ball of radius ρ_0. We represent this maximum loss as $\max_{||\epsilon||_2<\rho_0} H(y|x; \theta + \epsilon)$. Zeroth-order flatness is the difference between the above term and the entropy loss $H(y|x; \theta)$ as:

$$\mathcal{R}_0(x; H, \theta) = \max_{||\epsilon||_2<\rho_0} \left(H(y|x; \theta + \epsilon) - H(y|x; \theta)\right). \quad (2)$$

First-order flatness $\mathcal{R}_1(x; H, \theta)$ is based on sample x, the entropy loss H, and the pre-trained source model parameters θ as Eq. 3. We first search a weight perturbation ϵ applied to the parameters θ to maximize gradient normalization of the entropy loss $\|\nabla H\left(y|x; \theta + \epsilon\right)\|_2$, where ϵ is constrained within a Euclidean ball of radius ρ_1. We define first-order flatness as this maximum gradient normalization, scaled by the perturbation radius ρ_1, as follows:

$$\mathcal{R}_1(x, H, \theta) = \rho_1 \max_{||\epsilon||_2<\rho_1} \|\nabla H\left(y|x; \theta + \epsilon\right)\|_2. \quad (3)$$

We obtain the total loss $\mathcal{L}(x; \theta)$ by a summation of the entropy loss, the flatness constraints, and the parameter regularization as:

$$\begin{aligned}\mathcal{L}(x; \theta) = H(y|x; \theta) + \lambda \mathcal{R}_0(x; H, \theta) \\ + (1 - \lambda)\mathcal{R}_1(x, H, \theta) + \lambda_r \|\theta - \theta_0\|_2^2,\end{aligned} \quad (4)$$

where λ_r controls the extent of parameter regularization, λ is a hyperparameter balancing two flatness and θ_0 denotes initial parameters.

Optimization of Flatness Constraints with Taylor Expansion To optimize the zeroth-order flatness constraint, we follow (Foret et al. 2021) and adopt a Taylor expansion to approximate the objective in Eq. 2 to solve the maximum loss problem and facilitate optimization. Our motivation is that solving the original maximum loss problem in Eq. 2 is technically hard, as $H(y|x; \theta + \epsilon)$ is a high-dimensional nonlinear function towards a weight perturbation ϵ. We thus take the approximation by a Taylor expansion to simplify the maximum problem for zeroth-order flatness optimization.

$\nabla \mathcal{R}_0(x; H, \theta)$ denotes the approximate gradient of zeroth-order flatness as Eq. 6. We first approximate the objective function in Eq. 2 with a first-order Taylor expansion:

$$H(y|x; \theta + \epsilon) - H(y|x; \theta) \approx \epsilon^T \nabla_\theta H(y|x; \theta) \quad (5)$$

By substituting this Taylor expansion Eq. 5 into Eq. 2, we get an approximation of the optimal weight perturbation $\epsilon_0^* \approx \arg\max_{\epsilon, \|\epsilon\|_2<\rho_0} \epsilon^T \nabla_\theta H(y|x; \theta)$. We solve this problem through a classical dual norm problem to obtain a solution $\hat{\epsilon}_0$ as $\rho_0 \frac{\nabla(H(y|x;\theta))}{\|\nabla(H(y|x;\theta))\|_2}$. We substitute $\hat{\epsilon}_0$ into the original maximum loss problem in Eq. 2 and get an approximation of the gradient of zeroth-order flatness as:

$$\begin{aligned}\nabla \mathcal{R}_0(x; H, \theta) \approx \nabla H(y|x; \theta + \hat{\epsilon}_0) \\ \hat{\epsilon}_0 = \rho_0 \frac{\nabla(H(y|x; \theta))}{\|\nabla(H(y|x; \theta))\|_2}\end{aligned} \quad (6)$$

As for first-order flatness, we follow (Zhang et al. 2023a) to approximate its gradient by taking gradients in multiple steps. See details of the approximation in App. C.

Experiments

Experimental Settings

We use three target datasets: **ImageNet-C** (Hendrycks and Dietterich 2019), **ImageNet-R** (Hendrycks et al. 2021) and **ImageNet-Sketch (ImageNet-K)** (Wang et al. 2019) and adopt two noisy datasets: **SSB-hard (SH)** (Vaze et al. 2022) and **NINCO** (Bitterwolf, Mueller, and Hein 2023) to provide noisy samples injected into the target datasets. We use classification error rate as metric and see details about datasets, metrics, implementation in App. A and B.

We employ various TTA methods as baselines for comparison. *Source* uses the fixed pre-trained model. We divide the remaining baselines into three categories: **(1) Methods without sample selection:** *Test-time batch normalization (BN stats)* (Nado et al. 2020) updates the statistics in BN via test-batch statistics. *Pseudo-label (PL)* (Lee et al. 2013) updates the BN parameters by using pseudo labels. *Tent* (Wang et al. 2021) uses entropy minimization to updates parameters. *CoTTA* (Wang et al. 2022) leverages a teacher-student training framework with augmentation-averaged predictions and random parameter recovery. **(2) Methods with sample selection:** *RoTTA* (Yuan, Xie, and Li 2023) employs a teacher-student training method with a category-balanced sampling by considering timeliness and uncertainty. *OSTTA* (Lee et al. 2023) monitors varying in the predicted confidence to distinguish noisy samples for adaptation. *UniEnt* (Gao, Zhang, and Liu 2024) uses a Gaussian mixture model to distinguish noisy samples based on maximum cosine similarity between representation and classifier weights for adaptation. **(3) Methods with sample selection and flatness minimization:** *SAR* (Niu et al. 2023) selects low-entropy samples and conducts sharpness-aware entropy minimization. *SoTTA* (Gong et al. 2023) filters high confidence samples into a category-balanced buffer and optimizes sharpness-aware entropy.

Main Results

In Table 1 and 2, we conduct test-time adaptation experiments for all methods on three target datasets (ImageNet-C, ImageNet-R, and ImageNet-K) with two noisy datasets (SH, NINCO). Our method MoTTA outperforms all baselines in ImageNet-C (in average errors), Imagenet-R, and Imagenet-K under the noisy datasets, which validates the effectiveness of our model in dealing with noisy data streams at TTA. Specifically, in ImageNet-C, our method significantly outperforms state-of-the-art models, reducing the average error by 3.2% (from 56.6%) compared to SoTTA with SH and by 3.1% (from 57.3%) compared to UniEnt with NINCO. For the baselines, methods *without sample selection* (BN Stats, TENT, and CoTTA) achieve a better adaptation result than Source, as both SH and NINCO are near OOD datasets with a non-severe impact. Baselines *with sample selection* (OSTTA and UniEnt) improve TENT with additional sample selection to avoid noisy samples, leading to better results. Baselines *with sample selection and flatness minimization* (SAR and SoTTA) incorporate sample selection with the search for flat loss regions to improve performance. SoTTA outperforms the sample-selection baselines

		Noise			Blur				Weather				Digital				
Noisy type	Method	Gau.	Shot	Imp.	Def.	Gla.	Mot.	Zoom	Snow	Fro.	Fog	Brit.	Cont.	Elas.	Pix.	JPEG	Avg.
	source	97.0	96.3	97.4	82.1	90.3	85.3	77.5	83.4	76.9	76.0	40.9	94.6	83.5	79.1	67.4	81.8
	PL	98.7	98.6	98.5	96.0	97.9	94.6	91.4	91.6	95.6	85.5	70.9	98.5	89.5	79.3	81.9	91.2
	BN Stats	87.4	85.7	86.5	82.2	83.4	77.3	68.5	70.4	69.4	58.7	37.3	84.4	61.3	56.7	58.6	71.2
	TENT	86.1	81.5	79.0	69.4	74.0	62.7	55.9	57.2	65.1	45.3	34.2	71.9	51.4	43.4	48.3	61.7
	CoTTA	84.4	82.5	83.1	77.2	80.4	71.5	64.2	65.3	65.2	53.1	35.1	75.0	57.6	51.2	54.8	66.7
SH	RoTTA	90.0	87.4	88.5	77.9	85.3	76.9	67.4	62.0	65.1	50.6	**33.2**	93.2	56.1	50.3	52.3	69.1
	SAR	77.3	73.1	75.7	72.1	74.1	65.5	57.1	58.0	61.3	45.8	34.1	65.9	51.3	43.6	48.8	60.2
	OSTTA	79.5	76.8	77.2	72.3	74.9	66.5	58.0	58.8	61.6	46.8	34.2	74.9	52.2	45.5	49.3	61.9
	UniEnt	73.1	71.1	71.6	71.5	74.5	65.4	58.0	58.2	60.7	46.1	33.9	63.6	52.8	45.5	49.7	59.7
	SoTTA	75.8	70.6	67.8	67.9	68.6	57.1	51.7	51.1	62.8	42.1	33.6	66.0	45.9	41.0	46.2	56.6
	MoTTA (ours)	**65.5**	**63.0**	**63.6**	**66.5**	**67.3**	**55.5**	**49.7**	**50.2**	**58.4**	**41.3**	33.4	**57.2**	**44.5**	**39.9**	**45.4**	**53.4**
	source	97.0	96.3	97.4	82.1	90.3	85.3	77.5	83.4	76.9	76.0	40.9	94.6	83.5	79.1	67.4	81.8
	PL	98.5	98.3	98.5	96.7	98.4	95.4	90.7	93.0	95.6	84.4	73.7	96.5	90.6	83.7	84.7	91.9
	BN Stats	86.3	84.3	85.1	81.4	82.9	77.4	68.3	70.1	68.5	58.9	37.0	82.9	61.2	58.0	60.0	70.8
	TENT	87.4	81.1	82.1	72.0	80.6	65.3	56.7	58.3	65.5	46.2	34.6	69.1	50.9	44.8	49.7	63.0
	CoTTA	82.7	80.5	81.2	77.2	80.0	72.6	64.1	64.8	64.1	53.0	34.5	73.8	56.9	52.3	56.0	66.2
NINCO	RoTTA	82.2	78.4	78.9	75.7	80.2	71.5	63.0	59.3	62.2	47.8	**32.6**	78.3	52.4	46.4	52.4	64.1
	SAR	74.6	70.8	71.0	72.0	73.9	66.0	56.9	57.1	60.7	45.5	34.0	64.6	50.2	44.2	49.3	59.4
	OSTTA	77.4	74.9	75.1	72.3	76.2	68.1	58.3	59.4	61.7	47.3	34.3	72.8	51.5	46.7	50.7	61.8
	UniEnt	69.4	67.5	67.8	68.2	71.0	62.6	55.6	56.0	**59.3**	45.0	33.4	58.3	51.0	44.5	49.3	57.3
	SoTTA	74.2	68.5	69.3	69.4	71.8	58.6	52.7	52.4	61.5	43.3	34.4	71.0	46.7	42.1	47.5	57.6
	MoTTA (ours)	**65.9**	**63.9**	**64.0**	**67.8**	**68.3**	**56.7**	**50.4**	**51.2**	60.1	**42.2**	34.2	**56.3**	**45.2**	**40.7**	**46.6**	**54.2**

Table 1: Classification error rates (%) on ImageNet-C for 15 types of corruptions under two noisy data types: SH and NINCO. **Bold** numbers are the lowest error rates. Underline numbers are the second lowest error rates.

	ImageNet-R		**ImageNet-K**	
Method	SH	NINCO	SH	NINCO
Source	63.8	63.8	75.9	75.9
PL	86.8	87.7	94.4	95.2
BN Stats	62.1	62.0	73.3	73.1
TENT	59.7	60.5	70.6	70.5
CoTTA	60.9	60.2	71.0	70.2
RoTTA	61.7	61.2	71.7	70.4
SAR	59.2	59.0	69.0	68.6
OSTTA	58.5	58.8	68.5	68.9
UniEnt	59.3	58.5	69.1	68.1
SoTTA	57.6	58.4	67.5	67.5
MoTTA (ours)	**56.1**	**56.7**	**66.0**	**66.8**

Table 2: Classification error rates (%) on ImageNet-R and ImageNet-K with two noisy data sets: SH and NINCO.

on most datasets (5 out of 6).

Ablation Studies

In Table 3, we conduct ablation studies on ImageNet-R and ImageNet-K with SH and NINCO as noisy datasets by removing the proposed modules to validate the effectiveness of components. In row 2 (w.o. ODP), the removal of ODP-based filtering (Sec. 3.4) increases the error rates among all datasets, proving that ODP-based filtering improves noisy sample filtering for adaptation. We find that removing first-order flatness (w.o. $\mathcal{R}_1$) in parameter learning (degrade to sharpness-aware entropy minimization in SoTTA, Sec. 3.5) raises classification error rates on all datasets, showing that the first-order flatness can improve adaptation performance

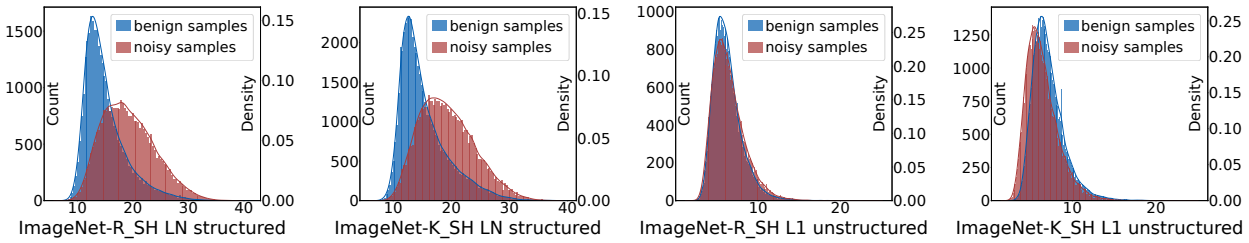
	ImageNet-R		**ImageNet-K**	
Noisy type	SH	NINCO	SH	NINCO
Ours	**56.1**	**56.7**	**66.0**	**66.8**
w.o. ODP	57.4	57.5	67.3	67.4
w.o. $\mathcal{R}_1$	56.5	56.8	66.5	67.3
w.o. $\mathcal{R}_r$	56.8	58.2	67.1	68.3
w.o. Conf.	57.2	57.9	68.1	68.1

Table 3: Classification error rate (%) on ImageNet-R and ImageNet-K with two noisy datasets (SH, NINCO) for variants of our model. Conf. denotes confidence-basd filtering.

on noisy data streams via the better flat search. After deleting parameter regularization (w.o. $\mathcal{R}_r$, Sec. 3.5), the classification error increases among all datasets, which validates the effectiveness of parameter regularization in mitigating misleading updates from noisy samples. Removing confidence-based filtering (w.o. Conf.) increases classification error rates for each dataset, proving the necessity of confidence-based filtering to improve the sample quality.

Study on Distribution of ODP Scores

To illustrate whether the ODP score is suitable to distinguish noisy samples, we collect the ODP scores on target samples versus those on noisy samples and draw the score distributions in Fig. 3. We adopt two kinds of combined datasets, ImageNet-R with SH and ImageNet-K with SH for ODP score calculation with structured pruning on output channels of convolutional layers (Conv. layers) in the initial pre-trained source model. As shown in Fig. 3. (a) and (b), the ODP scores show a bimodal distribution for noisy samples



(a) (b) (c) (d)

Figure 3: Distributions of ODP scores on two datasets (ImageNet-R with SH and ImageNet-K with SH) with two pruning strategies: LN structured pruning and L1 unstructured pruning.

and benign samples, which shows we can use the ODP score for distinguishing noisy samples based on a lack of clear boundaries between two peaks.

To explore different pruning strategies, we further collect the ODP scores of ImageNet-R with SH and ImageNet-K with SH under unstructured pruning (removing individual parameters with the smallest absolute weights). We plot the distributions of ODP score in Fig. 3. (c) and (d), respectively. We observe there is no obvious bimodal distribution under this pruning strategy for ODP-score. This phenomenon shows that we cannot use unstructured pruning for ODP score to distinguish noisy samples.

case 1 ratio	case 1 num.	case 2 ratio	case 2 num.	case 3 ratio	case 3 num.
44.6	43376	68.1	33943	64.2	21059

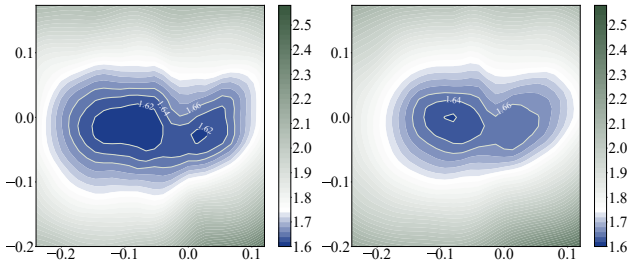
Table 4: Target sample ratio (%) in filtered samples and the number of filtered samples (num.) on ImageNet-R with SH using filtering. thresholds $\tau_{ODP} = 17, \tau c = 0.33$. Case 1: confidence-based filtering. Case 2: ODP-based filtering. Case 3: ODP-based filtering and confidence-based filtering.

Study on Effectiveness of ODP-Based Filtering

To validate the effectiveness of ODP-based filtering, we explore different filtering methods on ImageNet-R with SH via fixed pre-trained model in Table 4. We report the target sample ratio in the filtered samples and the number of filtered samples. We observe: (1) confidence-based filtering alone has a low ratio of target samples after filtering (case 1). (2) ODP-based filtering has a high target sample ratio (case 2), which indicates the effectiveness of this filtering in distinguishing noisy samples. (3) Confidence-based filtering significantly improves the target sample ratio by combining ODP-based filtering, which validates the effectiveness of our sample selection strategy in detecting noisy samples.

Study on Flatness of Loss Surface

To explore the effect of first-order flatness on the loss surface, we follow (Li et al. 2018) to visualize the entropy loss surface by adding perturbations to model weights in Fig. 4. We use the model after adaptation on ImageNet-K with SH, with two adaptation methods: MoTTA (ours, Fig 4.a), and



(a) MoTTA (b) MoTTA w.o. $\mathcal{R}_1$

Figure 4: Entropy loss surface on ImageNet-K with SH for MoTTA (ours) with or without first-order flatness $\mathcal{R}_1$.

ours without first-order flatness constraint (4.b). Our method shows a larger contour for the lowest loss compared to our method without first-order flatness constraint. This shows that first-order flatness constraint benefits the search for a flatter surface, which makes parameters more robust to noisy gradients caused by noisy samples.

Conclusion

We propose a test-time adaptation model to handle noisy data streams by involving noisy sample filtering. The model is based on model pruning and a combined flatness constraint searching for flat loss regions. Through output difference before and after applying model pruning, we can distinguish noisy samples from benign ones even when the prediction is incorrect under domain shifts. The combined flatness constraint forces both the worst case and the maximum gradient norms within a weight perturbation, which contributes to the resilience of parameters to deal with misleading gradients from noisy samples. The experiments show our method excels in test-time adaptation on noisy data streams.

Acknowledgments

This work is supported by the following fundings: Young Elite Scientist Sponsorship Program by CAST (2023QNRC001) under Grant No. YESS20230367, the National Natural Science Foundation of China under Grant No. 62306330, and Hong Kong Research Grants Council under Grant No. 16204920. We gratefully acknowledge the support of NVIDIA Corporation with the GPU resources.

References

Bendale, A.; and Boult, T. E. 2016. Towards Open Set Deep Networks. In *CVPR*.

Bitterwolf, J.; Mueller, M.; and Hein, M. 2023. In or Out? Fixing ImageNet Out-of-Distribution Detection Evaluation. In *ICLR-RTML Workshop*.

Boudiaf, M.; Mueller, R.; Ben Ayed, I.; and Bertinetto, L. 2022. Parameter-free Online Test-time Adaptation. In *CVPR*.

Bucci, S.; Loghmani, M. R.; and Tommasi, T. 2020. On the Effectiveness of Image Rotation for Open Set Domain Adaptation. In *ECCV*.

Chen, D.; Wang, D.; Darrell, T.; and Ebrahimi, S. 2022. Contrastive Test-Time Adaptation. In *CVPR*.

Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2021. Sharpness-aware Minimization for Efficiently Improving Generalization. In *ICLR*.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. S. 2016. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.*

Gao, Z.; Zhang, X.-Y.; and Liu, C.-L. 2024. Unified Entropy Optimization for Open-Set Test-Time Adaptation. In *CVPR*.

Gong, T.; Kim, Y.; Lee, T.; Chottananurak, S.; and Lee, S.-J. 2023. SoTTA: Robust Test-Time Adaptation on Noisy Data Streams. In *NeurIPS*.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; Song, D.; Steinhardt, J.; and Gilmer, J. 2021. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *ICCV*.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *ICLR*.

Hendrycks, D.; and Gimpel, K. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *ICLR*.

Iwasawa, Y.; and Matsuo, Y. 2021. Test-Time Classifier Adjustment Module for Model-Agnostic Domain Generalization. In *NeurIPS*.

Kim, H. E.; Cosa-Linan, A.; Santhanam, N.; Jannesari, M.; Maros, M. E.; and Ganslandt, T. 2022. Transfer learning for medical image classification: a literature review. *BMC medical imaging*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *PNAS*.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *NeurIPS*.

Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*.

Lee, J.; Das, D.; Choo, J.; and Choi, S. 2023. Towards open-set test-time adaptation utilizing the wisdom of crowds in entropy minimization. In *ICCV*.

Li, H.; Ding, L.; Fang, M.; and Tao, D. 2024. Revisiting Catastrophic Forgetting in Large Language Model Tuning. In *Findings of EMNLP*.

Li, H.; Xu, Z.; Taylor, G.; Studer, C.; and Goldstein, T. 2018. Visualizing the Loss Landscape of Neural Nets. In *NeurIPS*.

Liang, J.; He, R.; and Tan, T. 2024. A comprehensive survey on test-time adaptation under distribution shifts. *IJCV*.

Liang, S.; Li, Y.; and Srikant, R. 2018. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In *ICLR*.

Lim, H.; Kim, B.; Choo, J.; and Choi, S. 2023. TTN: A Domain-Shift Aware Batch Normalization in Test-Time Adaptation. In *ICLR*.

Liu, H.; Cao, Z.; Long, M.; Wang, J.; and Yang, Q. 2019. Separate to adapt: Open set domain adaptation via progressive separation. In *CVPR*.

Liu, J.; Yang, S.; Jia, P.; Zhang, R.; Lu, M.; Guo, Y.; Xue, W.; and Zhang, S. 2024. ViDA: Homeostatic Visual Domain Adapter for Continual Test Time Adaptation. In *ICLR*.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning transferable features with deep adaptation networks. In *ICML*.

Luo, Y.; Wang, Z.; Huang, Z.; and Baktashmotlagh, M. 2020. Progressive Graph Learning for Open-Set Domain Adaptation. In *PMLR*.

Nado, Z.; Padhy, S.; Sculley, D.; D'Amour, A.; Lakshminarayanan, B.; and Snoek, J. 2020. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv:2006.10963*.

Niu, S.; Miao, C.; Chen, G.; Wu, P.; and Zhao, P. 2024. Test-Time Model Adaptation with Only Forward Passes. In *ICML*.

Niu, S.; Wu, J.; Zhang, Y.; Chen, Y.; Zheng, S.; Zhao, P.; and Tan, M. 2022. Efficient Test-Time Model Adaptation without Forgetting. In *ICML*.

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

Saito, K.; Yamamoto, S.; Ushiku, Y.; and Harada, T. 2018. Open Set Domain Adaptation by Backpropagation. In *ECCV*.

Schneider, S.; Rusak, E.; Eck, L.; Bringmann, O.; Brendel, W.; and Bethge, M. 2020. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*.

Vaze, S.; Han, K.; Vedaldi, A.; and Zisserman, A. 2022. Open-Set Recognition: A Good Closed-Set Classifier is All You Need. In *ICLR*.

Wan, F.; Zhao, H.; Yang, X.; and Deng, C. 2024. Unveiling the Unknown: Unleashing the Power of Unknown to Known in Open-Set Source-Free Domain Adaptation. In *CVPR*.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B. A.; and Darrell, T. 2021. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *ICLR*.

Wang, H.; Ge, S.; Lipton, Z.; and Xing, E. P. 2019. Learning Robust Global Representations by Penalizing Local Predictive Power. In *NeurIPS*.

Wang, Q.; Fink, O.; Gool, L. V.; and Dai, D. 2022. Continual Test-Time Domain Adaptation. In *CVPR*.

Yang, H.; Wang, M.; Jiang, J.; and Zhou, Y. 2024a. Towards Test Time Adaptation via Calibrated Entropy Minimization. In *KDD*.

Yang, H.; Zuo, H.; Zhou, R.; Wang, M.; and Zhou, Y. 2024b. Towards Test Time Domain Adaptation via Negative Label Smoothing. *Neurocomputing*.

Yuan, L.; Xie, B.; and Li, S. 2023. Robust test-time adaptation in dynamic scenarios. In *CVPR*.

Yuan, Y.; Xu, B.; Hou, L.; Sun, F.; Shen, H.; and Cheng, X. 2024. TEA: Test-time Energy Adaptation. In *CVPR*.

Zhang, X.; Xu, R.; Yu, H.; Dong, Y.; Tian, P.; and Cui, P. 2023a. Flatness-aware minimization for domain generalization. In *ICCV*.

Zhang, X.; Xu, R.; Yu, H.; Zou, H.; and Cui, P. 2023b. Gradient norm aware minimization seeks first-order flatness and improves generalization. In *CVPR*.