Layer-wise Parameter Robustness for Continual Test-time Adaptation

Haoyu Xiong[1,†], Qiuxia Yang[1,†], Chengchao Wang[1], Tianze Zhong[1], Zhengpeng Zhao[1], and Yuanyuan Pu[1,2,‡]
[1] School of Information Science and Engineering, Yunnan University, Kunming, China
[2] The Universities Key Laboratory of Internet of Things Technology and Application in Yunnan, Kunming, China
Email: xionghaoyu@stu.ynu.edu.cn, yuanyuanpu@ynu.edu.cn

***Abstract*—Since inevitable distribution shifts are encountered during test time in practice, test-time adaptation (TTA) presents a promising solution by recalibrating the model online using only an unlabeled test data stream. However, TTA often suffers from issues such as catastrophic forgetting caused by continuously changing environments, as it relies on self-training. Contemporary solutions attempt to mitigate this by anchoring TTA to a static source model, such as stochastic parameter restoration or periodic parameter reset, which restrict model flexibility. Moreover, different layers may exhibit varying sensitivities to distribution shifts, sometimes even showing opposite shift trends, yet prior methods treat all layers homogeneously. Motivated by this, we propose a layer-wise parameter robustness method that autonomously identifies important parameters in different layers for selective weighting by measuring the sharpness of parameter surface. Further in-depth experiments on various benchmarks demonstrate the robustness and effectiveness of our proposed method. Our code is available at https://github.com/ioslide/prda_tta.**

***Index Terms*—Test-time Adaptation, Domain shift**

I. Introduction

Deep Neural Networks (DNNs) have demonstrated groundbreaking success in various fields [1], [2], but they rely on a critical assumption that the source and target domains share the same distribution. Unfortunately, even the most advanced DNNs often violate this assumption [3]. To address model performance degradation, prior studies have primarily focused on improving model robustness and generalization ability during training, such as domain adaptation [4]. Despite this progress, it remains unrealistic to cover a wide and unknown range of shifts. Moreover, the source data is often inaccessible during test time due to privacy or legal restrictions. Since test data can provide insights into the current distribution shift, test-time adaptation (TTA) [5] presents a promising solution, recalibrating models in an online manner using unlabeled test data without requiring additional labeled data. Undoubtedly, TTA considers a more challenging but practical setting and bypasses numerous restrictions, such as privacy protection.

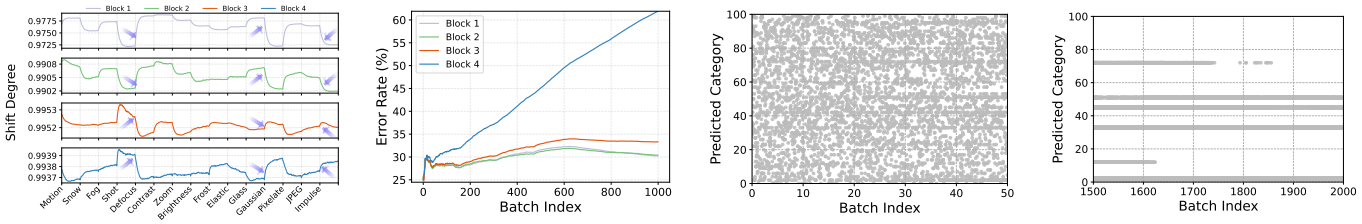
Nevertheless, the journey of TTA is fraught with obstacles, particularly when the target distribution undergoes continual or even unexpected changes over time. For example, an autonomous vehicle's sensors might encounter overexposure or weather changes (sunny to rainy, day to night). Moreover, these changes can be abrupt and dramatic. However, as TTA is rooted in the self-training paradigm, it is prone to causing pseudo-labels to become noisier and miscalibrated over time, leading to error accumulation and catastrophic forgetting. Under such circumstances, previous TTA methods [5] may fail or even degrade performance, ultimately resulting in trivial solutions (see Fig. 1d). Thus, it is crucial to consider TTA in a continual manner, *i.e.*, continual test-time adaptation (CTTA) [6].

Such a setting is particularly challenging, as maintaining source domain knowledge becomes increasingly difficult in the long term. Some contemporary TTA methods mitigate this by bundling adaptation with a frozen source model, such as stochastic parameter restoration [6], weight averaging [7], and periodic parameter resets [8]. However, these approaches hinder flexibility. Moreover, different layers may exhibit varying sensitivities to distribution shifts, or even show opposite shift trends within the same domain (see Fig. 1a), such as *Defocus*→*Contrast*. Existing methods typically treat all layers homogeneously, which can trap the model in a high-loss error pool [9], leading to suboptimal adaptation (see Fig. 1b). We hypothesize that layer-wise fine-tuning could further enhance adaptability to continually changing distribution shifts.

Inspired by this, we propose a layer-wise parameter robustness TTA method for CTTA by autonomously identifying important parameters in each layer for selective weighting. The goal of our method is to utilize the homeostatic knowledge from the source pre-trained model to prevent catastrophic forgetting while preserving essential knowledge in the current model. To achieve this, we approximate the Hessian matrix using the Fisher Information Matrix (FIM) to quantify the sharpness of each layer. By assessing sharpness levels, we can determine the extent of parameter changes, enabling the update or preservation of critical parameters for each layer. Additionally, we introduce a dual-view alignment strategy to prevent FIM bias and mitigate error accumulation from noisy pseudo-labels. This simple method withstands extensive experiments and ablation studies on various TTA benchmarks, confirming its effectiveness and robustness. *Our key contributions are summarized as follows:*

- By analyzing the distribution shift of each layer, we reveal that different layers may exhibit opposite or significantly different shift trends during TTA process.
- We motivate and propose a layer-wise parameter robustness method for CTTA to autonomously identify

‡ Corresponding author
† Equal contribution.



(a) Shift Degree (b) Evolution of Error Rate (c) No Collapse (Early) (d) Model Collapse (Late)

Fig. 1. (a) investigates the shift degree measured by the Gaussian kernel in each layer/block on ImageNet-C with ResNet-50 under CTTA. The position indicated by the arrow shows that different layers may exhibit opposite or significantly different shift trends. (b) illustrates how the error rate evolves in each layer. (c-d) record the prediction categories during the adaptation process on CIFAR100-C, revealing that error accumulation is difficult to detect early and leads to model collapse in the later stage, *i.e.*, assigning a few fixed category labels to all samples.

important parameters in each layer for selective weighting.
- Extensive experiments conducted on various TTA benchmarks, including CIFAR100-C, ImageNet-C, and ImageNet-3DCC, demonstrate the superiority of our method, which is also straightforward to implement.

II. Related Work

Domain Adaptation aims to enhance target domain performance by leveraging knowledge from both the source and target domains. To improve generalization performance on the target domain, various approaches focus on learning domain-invariant representations, employing techniques like contrastive learning [10], and domain discriminators [11]. To avoid reliance on source domain data, some research focuses on domain generation [12]. However, these methods typically rely on the entire target domain dataset and are executed offline, which has prompted research into TTA.

Test-time Adaptation (TTA) aims to enhance model performance during test time, enabling the model to effectively adapt to unknown target domains using only an unlabeled test data stream. Since the test data provide insights into the distribution shifts, leveraging batch normalization statistics [13] from test samples, TTA achieves stable improvements with only one forward pass. Tent [5] extends this by adding a backward pass to update batch normalization parameters via entropy minimization. This method has inspired further studies on robustness under continual distribution shifts [6], presenting challenges to traditional TTA methods.

Two key challenges in this context are catastrophic forgetting and error accumulation. To address error accumulation, methods include using mean-teacher models for better pseudo-labels and minimizing entropy of reliable samples [8]. For catastrophic forgetting, approaches like CoTTA [6] stochastically restore some pre-trained parameters, and RMT [14] uses contrastive loss with source prototypes. However, these methods neglect layer-wise anisotropy to distribution shifts, limiting adaptability.

III. Methodology

A. Problem Definition

Without loss of generality, let the labeled data from the source distribution $\mathcal{P}^{\mathrm{S}}$ be $\mathcal{D}^{\mathrm{S}} = \{(x^{\mathrm{S}}, y^{\mathrm{S}})\}$, and let the unlabeled data from the target distribution $\mathcal{P}_t^{\mathrm{T}}$ at each time step $t \in \{1, 2, \ldots, T\}$ be $\mathcal{D}_t^{\mathrm{T}} = \{x_t^{\mathrm{T}} \sim \mathcal{P}_t^{\mathrm{T}}\}$, where $\mathcal{P}^{\mathrm{S}} \neq \mathcal{P}^{\mathrm{T}}$. Given a model f_{θ_0} with parameters $\theta_0 = \{\theta_i^l\}_{i=1}^{|\theta_0|}$ pre-trained on $\mathcal{D}^{\mathrm{S}} \sim \mathcal{P}^{\mathrm{S}}$, the model suffers significant performance degradation on $\mathcal{P}^{\mathrm{T}}$ due to continual distribution shifts over time, *i.e.*, $\mathcal{P}_1^{\mathrm{T}}, \mathcal{P}_2^{\mathrm{T}}, \ldots, \mathcal{P}_T^{\mathrm{T}}$.

Instead, test-time adaptation (TTA) [5] presents a promising solution, recalibrating the model parameters in an online manner using only the current test data. At each time step t, given the TTA objective function $\mathcal{L}(x_t, \theta_t)$, the iterative optimization of each layer's parameters θ_t^l is as follows:

$$\begin{aligned} \theta_{t+1}^l &:= \theta_t^l - \eta \Delta_{\theta_t} \mathcal{L}(x_t, \theta_t), \\ \text{where} \quad \Delta\theta_t^l &= \frac{\partial \mathcal{L}(x_t, \theta_t)}{\partial \theta_t^l}. \end{aligned} \tag{1}$$

Here, η is the learning rate, and $\mathcal{L}(\cdot)$ can be formulated as an entropy minimization [5] problem or other variants [8], [15].

B. Layer-wise Parameter Robustness

Assuming the adaptation to t target domains has been completed at time step t, resulting in t composite models with different parameters $(\theta_1, \ldots, \theta_t)$, represented by $\tilde{\theta}_t$, ensuring effective performance across all target domains without catastrophic forgetting. If $\tilde{\theta}_t$ is assumed to follow an isotropic Gaussian distribution, then parameter averaging is conducted on the implicit premise that all weights are equally important to the observed data x_t, *i.e.*, $\tilde{\theta}_t = -\nabla \sum_{i=1}^{T} \mathcal{L}(x_t, \theta_i)$. To simplify, let us consider two models with parameters θ_{t-1} and θ_t, respectively. The parameter averaging is as follows:

$$\tilde{\theta}_t = -\mu \nabla \mathcal{L}(x_t, \theta_{t-1}) - (1-\mu) \nabla \mathcal{L}(x_x, \theta_t), \tag{2}$$

where μ is a trade-off parameter.

However, continuous distribution shifts may invalidate this assumption, pushing the model into distant parameter space regions and hindering parameter averaging. As section I and Fig. 1a show, different layers can exhibit distinct, even opposing, distribution shifts within the same target domain.

To address these issues, we propose a layer-wise parameter robustness method that autonomously identifies the important parameters of each layer for selective weighting. We first

quantify parameter importance by calculating the second-order derivative using the Hessian matrix of the layer parameters.

However, the complexity of the Hessian matrix is proportional to the square of the number of parameters, which may be infeasible for large models. To reduce computational overhead, we use the Fisher Information Matrix (FIM) to measure the sharpness of each layer's parameters. The FIM is the expectation of the Hessian or the second-order derivative of the log-likelihood with respect to the parameters, encapsulating the sharpness of each layer's parameters θ_t^l [16]. When the associated FIM value of a specific layer is extremely low, it indicates that the layer's parameters are close to optimal. We then obtain the layer-wise FIM $\mathcal{I}_{\theta_t^l}$ by calculating the second-order derivative of the log-likelihood with respect to the parameters, as follows:

$$\mathcal{I}_{\theta_t^l} = \mathbb{E}_{x_t \sim \mathcal{D}^T} \left[\nabla_{\theta_t^l} \log f_{\theta_t}(x_t) \nabla_{\theta_t^l} \log f_{\theta_t}(x_t)^T \right]. \quad (3)$$

Note that the FIM $\mathcal{I}_{\theta_t^l}$ is calculated for each layer θ_t^l of the model f_{θ_t} with respect to the test data x_t.

To prevent catastrophic forgetting and improve generalization, we bind the adapted model by integrating a small amount of crucial knowledge from the source model (with parameters θ_0). Inspired by [7], [17], the merged parameter $\tilde{\theta}_t$ has the following closed-form solution:

$$\tilde{\theta}_t = \sum_{l}^{|\theta_t|} \mathbf{1}\left[l \in \text{BN}\right] \cdot \frac{\mu \mathcal{I}_{\theta_t^l} \theta_t^l + (1-\mu)\mathcal{I}_{\theta_0^l} \theta_0^l}{\mu \mathcal{I}_{\theta_t^l} + (1-\mu)\mathcal{I}_{\theta_0^l}}, \quad (4)$$

where $\mu \in (0,1)$ is a tradeoff parameter, and $\mathbf{1}$ indicates whether the layer is a batch normalization layer. Since the source domain is inaccessible, the computation of $\mathcal{I}_{\theta_0}$ is approximated by the first iteration of the FIM $\mathcal{I}_{\theta_t}$.

C. Overall Update

Dual-view Alignment (DA). Despite the effectiveness of the 'layer-wise parameter robustness' component in Eq. (4), error accumulation is unavoidable in TTA's self-training paradigm, potentially biasing the FIM. To address this problem, we introduce a dual-view alignment loss to align with the source domain and mitigate the tendency of pseudo-labels to become noisier. The dual-view alignment loss is formulated as:

$$\mathcal{L}_{\text{DA}} = -\lambda_1 \mathbb{E}_{x_t}\left[p_{\theta_t}(x_t) \log p_{\theta_0}(x_t)\right] - \lambda_2 \mathbb{E}_{\hat{x}_t}\left[p_{\theta_t}(x_t) \log p_{\theta_t}(\hat{x}_t)\right]. \quad (5)$$

Here, $\hat{x}_t$ represents augmented data generated through a combination of data augmentations [18], such as *color jitter and Gaussian noise*. $p_{\theta_0}(x_t)$ denotes the softmax probabilities of the anchor model f_{θ_0} for x_t, and λ_* is a trade-off parameter. The anchor model parameters θ_0 remain fixed during TTA.

Loss function. Building upon the negative log-likelihood, our overall objective function is formulated as:

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{Shannon}} + \mathcal{L}_{\text{DA}}, \quad (6)$$

where $\mathcal{L}_{\text{Shannon}}$ is the Shannon entropy from Tent [5].

Layer-wise Optimization. According to the previously mentioned layer-wise parameter robustness method in Section III-B, the layer-wise optimization process with the gradient $\Delta\theta_t^l$ with respect to the loss function $\mathcal{L}_{\text{overall}}$ is as follows:

$$\begin{aligned} \theta_t^l &:= \theta_t^l - \eta * \Delta\theta_t^l, \\ \theta_{t+1}^l &:= \tilde{\theta}_t^l \xleftarrow{\text{via } Eq.\ (4)} \theta_t^l. \end{aligned} \quad (7)$$

This allows us to separately consider the importance of each layer's parameters, enabling more flexible test-time adaptation. Algorithm 1 briefly summarizes our method.

Algorithm 1: The proposed method.

Require: Pretrained parameter θ_0, learning rate η;
Initialize $\theta_t = \theta_0$;
for each time step t **do**
  Sample a mini-batch $x_t \in \mathcal{D}_t^{\text{T}}$;
  for each layer l **do**
    Compute gradient $\Delta\theta_t^l = \frac{\partial \mathcal{L}_{\text{overall}}}{\partial \theta_t^l}$ with objective function in Eq. (6);
    Update layer-wise FIM $\mathcal{I}_{\theta_t^l}$ and $\mathcal{I}_{\theta_0^l}$ according to Eq. (3);
    Update merged parameter $\tilde{\theta}_t$ in Eq. (4);
    Update parameter θ_t^l to $\tilde{\theta}_{t+1}^l$ in Eq. (7);

Result: Predictions $f_{\theta_t}(x_t)$.

IV. Experiments

A. Setup

Benchmarks. We evaluate our methods on several challenging TTA benchmarks, including CIFAR100-C, ImageNet-C [21], and ImageNet-3DCC [22]. The first two datasets consist of 15 different 2D synthetic corruptions, covering noise, blur, weather, and digital categories. The ImageNet-3DCC dataset consists of 12 different 3D real-world corruptions, which incorporate 3D information to generate corruptions consistent with scene geometry. We report the error rate following [6], [14].

Baselines and Competitors. We primarily compare our approach against other **continual TTA** methods that use arbitrary off-the-shelf pre-trained models, including *CoTTA* [6], *RoTTA* [19], and *TRIBE* [20], as well as **source-free TTA** methods, including *Tent* [5], *BN Adapt* [13], *SAR* [8], and the energy-based *TEA* [15]. Additionally, we also consider the **non-source-free** method *RMT* [14], although it is **unfair** to compare it with our method since the non-source-free setting violates the entire paradigm of fully TTA by accessing the source domain. *Source Only* indicates the pre-trained model directly evaluated on the target domain without adaptation.

Implementation Details. Following *RobustBench* [23], we use ResNeXt-29 [24] for CIFAR100-C and ResNet-50 [2] for ImageNet-C and ImageNet-3DCC. For optimization, we follow [5]: SGD optimizer (learning rate = 0.00025, momentum = 0.9) for ImageNet* family, and Adam optimizer (learning rate = 0.001, $\beta = 0.9$) for CIFAR100-C. For fair comparison, we set

TABLE I

ONLINE CLASSIFICATION ERROR RATE (%) UNDER CTTA. THE **BOLD** NUMBERS INDICATE THE BEST PERFORMANCE, AND THE RED NUMBERS INDICATE THE RESULTS THAT ARE LOWER THAN THE *Source only*.

(a) CIFAR100 → CIFAR100-C, AND IMAGENET → IMAGENET-C.

	Time				t →															
	Method	REF	Source-free	Updates	*Motion*	*Snow*	*Fog*	*Shot*	*Defocus*	*Contrast*	*Zoom*	*Brightness*	*Frost*	*Elastic*	*Glass*	*Gaussian*	*Pixelate*	*JPEG*	*Impulse*	Mean
CIFAR100-C	Source only	/	✓	✗	30.79	39.46	50.28	67.99	29.33	55.03	28.78	29.50	45.80	37.21	54.14	72.97	74.69	41.23	39.37	46.44
	BN Adapt [13]	ICML'15	✓	✗	30.27	35.84	42.40	41.82	28.49	30.94	28.87	27.38	35.41	36.38	42.66	43.29	33.84	42.01	43.74	36.22
	Tent [5]	ICLR'21	✓	✓	28.68	35.87	47.43	71.48	79.72	89.87	94.35	96.36	97.23	97.24	97.40	97.44	97.57	97.79	97.70	81.74
	CoTTA [6]	CVPR'22	✓	✓	30.70	36.61	45.54	39.90	29.43	32.60	29.57	27.97	36.65	38.78	42.76	43.60	34.25	41.18	45.98	37.03
	SAR [8]	ICLR'23	✓	✓	29.64	33.62	37.20	36.01	**26.00**	27.74	25.55	24.64	30.42	**32.05**	36.57	**36.24**	28.94	37.21	38.42	32.02
	RoTTA [19]	CVPR'23	✓	✓	32.41	37.40	40.41	41.01	28.04	34.64	26.21	24.23	29.80	32.34	36.37	35.77	28.80	36.28	40.26	33.60
	RMT [14]	CVPR'23	✗	✓	27.93	31.25	34.76	34.52	26.17	27.76	26.22	24.38	30.66	32.23	36.12	34.96	28.76	36.12	37.97	31.32
	TEA [15]	CVPR'24	✓	✓	45.77	91.87	95.39	96.03	96.29	96.38	96.30	96.34	96.52	96.42	96.52	96.49	96.53	96.62	96.33	92.65
	TRIBE [20]	AAAI'24	✓	✓	28.62	32.33	37.57	39.38	28.08	28.18	26.41	24.91	31.60	33.92	38.08	40.35	31.79	38.21	38.72	33.21
	Ours	/	✓	✓	**27.54**	**30.25**	**33.30**	**34.49**	26.54	**27.61**	**25.53**	**24.14**	**30.32**	33.23	**36.08**	36.79	**28.69**	**35.20**	**33.50**	**30.88**
ImageNet-C	Source only	/	✓	✗	96.46	96.72	96.32	97.08	97.90	99.88	94.16	70.54	92.92	95.16	97.48	97.10	87.12	76.02	97.28	92.81
	BN Adapt [13]	ICML'15	✓	✗	76.02	66.56	56.30	83.72	87.18	90.86	64.04	36.56	69.82	56.66	85.90	83.98	52.22	59.30	83.42	70.17
	Tent [5]	ICLR'21	✓	✓	72.62	61.00	48.62	76.20	77.34	75.94	**56.08**	37.58	63.26	50.46	73.32	72.96	48.52	54.36	69.46	62.51
	CoTTA [6]	CVPR'22	✓	✓	75.94	66.22	55.75	82.41	85.75	89.24	61.91	35.79	67.62	53.67	82.87	80.07	49.39	56.16	79.44	68.15
	SAR [8]	ICLR'23	✓	✓	72.48	61.08	49.02	76.36	77.06	75.36	56.12	36.86	63.20	49.68	71.92	71.88	47.44	52.22	67.80	61.90
	RoTTA [19]	CVPR'23	✓	✓	80.80	70.68	59.58	85.32	88.10	84.08	62.82	36.26	66.24	53.78	77.32	82.80	50.70	52.50	78.32	68.62
	RMT [14]	CVPR'23	✗	✓	69.02	58.83	47.11	74.59	78.01	78.53	55.72	**35.33**	60.81	48.85	73.99	70.61	44.95	49.85	67.67	60.92
	TEA [15]	CVPR'24	✓	✓	66.88	85.28	97.66	99.38	99.56	99.64	99.62	99.60	99.76	99.72	99.68	99.66	99.60	99.66	99.66	96.36
	TRIBE [20]	AAAI'24	✓	✓	72.72	60.36	50.51	77.83	79.29	84.59	59.07	36.47	62.37	50.36	74.54	73.35	49.53	52.06	71.08	63.61
	Ours	/	✓	✓	**66.36**	**56.30**	**46.72**	**70.30**	**73.96**	**71.90**	56.82	36.42	**60.18**	**47.96**	**69.18**	**67.94**	**44.94**	**48.28**	**64.18**	**58.76**

(b) IMAGENET → IMAGENET-3DCC.

	Time				t →													
	Method	REF	Source-free	Updates	*Flash*	*H265 abr*	*Far focus*	*Bit error*	*Low light*	*Z motion blur*	*Fog 3d*	*Near focus*	*ISO noise*	*H265 crf*	*XY motion blur*	*Color quant*	Mean
ImageNet-3DCC	Source only	/	✓	✗	90.38	71.66	79.24	89.12	94.80	90.18	94.04	74.34	94.80	64.36	93.88	90.98	85.65
	BN Adapt [13]	ICML'15	✓	✗	82.24	48.50	56.52	75.10	61.12	69.44	79.68	47.48	76.34	36.26	80.60	71.44	65.39
	Tent [5]	ICLR'21	✓	✓	80.48	45.18	52.64	71.62	55.82	61.64	73.76	44.14	64.76	35.94	73.88	66.90	60.56
	CoTTA [6]	CVPR'22	✓	✓	82.24	48.48	56.52	75.10	61.12	69.46	79.62	47.42	76.46	36.22	80.64	71.42	65.39
	SAR [8]	ICLR'23	✓	✓	80.02	45.60	53.62	72.40	56.34	62.66	71.74	44.30	65.10	35.26	73.48	66.72	60.60
	RoTTA [19]	CVPR'23	✓	✓	80.40	46.42	55.74	72.22	66.10	72.22	**69.72**	44.90	75.04	38.32	77.20	66.50	63.73
	RMT [14]	CVPR'23	✗	✓	78.46	43.52	53.30	**70.32**	52.82	61.66	71.82	**42.92**	62.72	34.65	71.64	**64.11**	59.00
	TEA [15]	CVPR'24	✓	✓	**77.16**	73.50	95.70	99.54	99.60	99.68	99.80	99.70	99.66	99.62	99.68	99.70	95.28
	TRIBE [20]	AAAI'24	✓	✓	77.89	44.49	52.71	71.16	57.30	62.92	70.04	43.70	65.09	36.22	73.89	65.79	60.10
	Ours	/	✓	✓	78.24	**42.76**	**52.20**	71.01	**51.74**	**60.34**	72.30	43.34	**61.84**	**34.66**	**71.26**	**65.14**	**58.73**

the batch size to 64 and adhere to the official implementations of other TTA methods, unless otherwise specified.

B. Continual Test-Time Adaptation

Experimental Settings. We evaluate the performance of our method under the continual TTA (CTTA) [6] setting, where the corruption sequence consists of all 15 corruption domains in order, with the highest severity level set to 5.

$$\dots \underbrace{\mathcal{D}^{\mathrm{T}-1}}_{\mathcal{P}^{\mathrm{T}-1}} \xrightarrow[\text{Shift}]{\text{Domain}} \underbrace{\mathcal{D}^{\mathrm{T}}}_{\mathcal{P}^{\mathrm{T}}} \xrightarrow[\text{Shift}]{\text{Domain}} \underbrace{\mathcal{D}^{\mathrm{T}+1}}_{\mathcal{P}^{\mathrm{T}+1}} \dots \tag{8}$$

As shown in Eq. (8), the target domain distribution $\mathcal{P}^{\mathrm{T}}$ changes continuously and unpredictably over time.

Experimental Results. Table I comprehensively details the results of each TTA method across various datasets under the continual TTA setting. Overall, most methods contribute positively to model performance, emphasizing the necessity of test-time adaptation. Upon closer inspection, TEA [15] and Tent [5] exhibit commendable performance in the initial target domain, which can be described as a single static target domain. Unexpectedly, they fail to effectively address the problem of error accumulation, ultimately leading to model failure (*i.e.*, error rate $\geq 90\%$). Although the suboptimal method RMT [14] successfully mitigates error accumulation, it introduces the assumption of an accessible source domain. Furthermore, RMT [14] and TRIBE [20] still face a heavy computational burden (see Fig. 5a) due to the complexity of their frameworks (*e.g.*, mean-teacher and source prototypes). Notably, our method consistently achieves the lowest error rate across all datasets, even surpassing parameter-rich non-source-free methods.

C. Cross-domain Generalization.

Experimental Settings. Generalization is often an underestimated topic in TTA. To evaluate this, we adapt the model on a single domain and then evaluate it across all other 14 ImageNet-C target domains (see Eq. (9)).

$$\underbrace{\mathcal{D}^{\text{Snow}}}_{\text{Adaptation Domain}} \xrightarrow{\text{Evaluate}} \underbrace{\{\mathcal{D}^{\text{Fog}}, \cdots, \mathcal{D}^{\text{Impulse}}\}}_{\text{Target Domain}} \tag{9}$$

Experimental Results. As TTA relies on self-training, this may risk "*parameter over-specialization*" on narrow distributions, limiting generalization to unseen domains. As shown in Fig. 2, over-adaptation arises when weakly correlated features are pushed into the shared feature space, such as outdated snapshots in RoTTA's memory bank. In contrast, our method consistently outperforms in generalization.

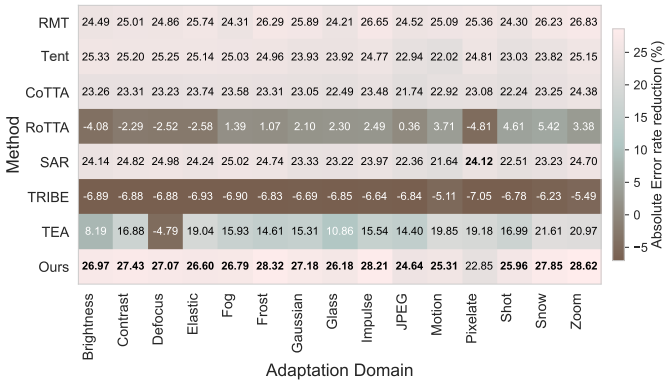


Fig. 2. Each cell at (i, j) denotes the average absolute error rate reduction of the i-th method adapted on the j-th domain (*e.g.*, Snow) from ImageNet-C (severity level 5) and evaluated across the remaining 14 domains (*e.g.*, Fog, $\cdots$, Impulse).

D. Ablation Study and Further Analysis

Component Analysis. To evaluate each component's effectiveness, we created two variants: (1) Ours w/o PR, which updates model parameters without the ensemble mechanism in Eq. (4); (2) Ours w/o DA, where the loss function degrades to using only Shannon entropy. As shown in Table II, both variants exhibit significant performance degradation, highlighting the importance of both components in our method.

TABLE II
EFFECTS OF COMPONENTS IN OUR METHOD ON IMAGENET-C UNDER CTTA. "PR" AND "DA" DENOTE EQ. (4) AND EQ. (5), RESPECTIVELY.

Method	CIFAR100-C	ImageNet-C	ImageNet-3DCC	Mean
Ours w/o PR	31.95	60.31	60.32	50.86
Ours w/o DA	32.59	62.75	61.57	52.30
Ours	**30.88**	**58.76**	**58.73**	**49.46**

Analysis of Hyperparameter. μ and λ_* control the proportion of source model knowledge in Eq. (4) and the trade-off in Eq. (5). As shown in Fig. 3a, incorporating a small amount of source domain knowledge effectively mitigates catastrophic forgetting. Furthermore, despite λ_1 and λ_2 spanning multiple orders of magnitude in Fig. 3b, performance remains stable within a narrow range near the diagonal, highlighting the importance of dual-view alignment in Eq. (5).

Analysis of Distribution Shifts with Different Orders. Since arbitrary orders of distribution changes may exist in reality, we further conduct experiments on different sequences of distribution changes. As shown in Fig. 4, our method effectively handles various distribution orders, including traditional independently sampled test data streams, making it a comprehensive method for real-world applications.

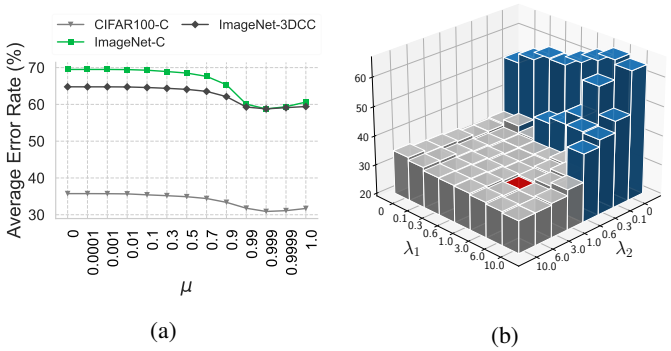


Fig. 3. (a) shows the influence of μ in Eq. (4). (b) investigates the sensitivity of λ_* in Eq. (5) on CIFAR100-C under CTTA.

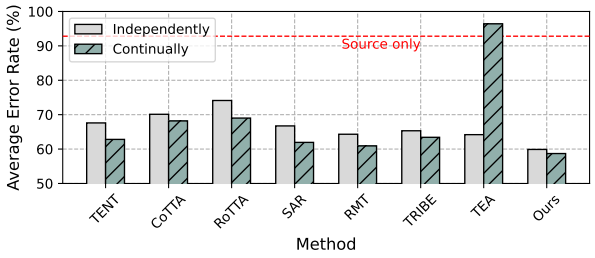


Fig. 4. Average error rate (%) for six different orders of distribution shifts with independently and continually sampled test streams on ImageNet-C.

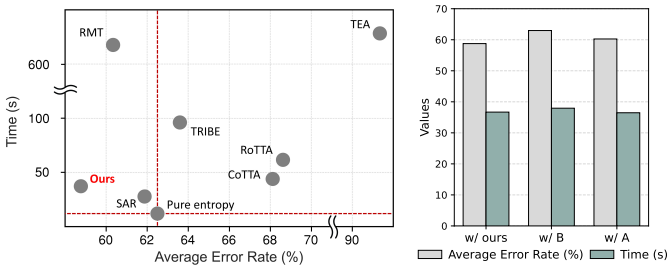
Robustness Across Model Architectures. To validate the 'model-agnostic' property, we used several distinct encoder configurations for evaluation, including the recent Vision Transformer (ViT) [1]. The results in table III again demonstrate the effectiveness of our method across various model architectures.

TABLE III
AVERAGE ERROR RATE (%) OF DIFFERENT MODEL ARCHITECTURES ON IMAGENET-C UNDER CTTA.

Method	ResNet-18 [2]	WideResNet-50 [25]	ResNext-50 [24]	ResNet-101 [2]	ViT-B [1]
Source only	94.46	78.58	78.49	90.13	59.51
BN Adapt	74.22	66.14	67.06	68.87	59.51
Tent	68.88	56.53	57.59	60.37	54.56
CoTTA	73.00	62.71	64.43	62.61	59.69
SAR	68.22	56.62	57.14	57.57	54.60
RoTTA	75.28	62.08	63.94	64.38	N/A
TEA	94.46	93.58	95.49	92.13	97.28
TRIBE	70.13	58.76	60.49	72.11	N/A
Ours	**58.76**	**54.17**	**55.57**	**56.09**	**52.13**

Computation Cost Measured by GPU Run Time. From Fig. 5a, RMT [14] introduces significant computational overhead due to pre-heating, while energy-based TEA [15] faces similar challenges. In contrast, since we use the diagonal FIM and update only the model's normalization layer, the additional computational overhead is negligible. Overall, our method achieves a computational speed second only to the efficiency-specialized SAR, reaching 0.007 s per image, while maintaining the lowest error rate.

Comparison with Other Ensemble Methods. Fig. 5b further investigates the effectiveness of our layer-wise parameter robustness in Eq. (4) compared to similar methods. Due to the different activation values caused by continuous distribution shifts, *Gradient Magnitude-based Weighting* (B) cannot effectively improve the model. In contrast, our method significantly reduces the error rate to 58.76% with only a small overhead compared to *Parameter Averaging* (A).



(a) (b)

Fig. 5. (a) Time (s) vs. Error rate (%). The y-axis and x-axis represent the GPU runtime and the average error rate on ImageNet-C under CTTA, respectively. (b) Comparison with parameter ensemble variants on ImageNet-C under CTTA. We report the GPU runtime for 5,000 images over 100 runs on a single A40 GPU.

V. Conclusion

In this work, we propose a layer-wise parameter robustness method to boost continual test-time adaptation performance. To achieve this, we first empirically analyze the sensitivity of different layers to distribution shifts, revealing that different layers may exhibit opposite shift trends within the same target domain. Subsequently, we leverage FIM to autonomously identify important parameters in different layers and perform selective weighting, mitigating issues of catastrophic forgetting and error accumulation. Our method has withstood extensive experiments, and promising results demonstrate its effectiveness and robustness.

Acknowledgment

This work is supported in part by National Natural Science Foundation of China under Grants 61271361, 61761046, 62162068, 52102382 and 62362070; in part by the Key Project of Applied Basic Research Program of Yunnan Provincial Department of Science and Technology under Grants 202001BB050043, 202401AS070149; in part by the Major Science and Technology Special Project in Yunnan Province under Grant 202302AF080006; in part by the Yunnan Key Laboratory of low-light Night Vision Technology and Intelligent Visual Navigation under Grant 202449CE340004.

References

[1] 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 *ICLR*, 2021. 1, 5

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778. 1, 3, 5

[3] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al., "Wilds: A benchmark of in-the-wild distribution shifts," in *ICML*, 2021, pp. 5637–5664. 1

[4] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy, "Domain generalization: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 1

[5] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *ICLR*, 2021. 1, 2, 3, 4

[6] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai, "Continual test-time domain adaptation," in *CVPR*, 2022, pp. 7201–7211. 1, 2, 3, 4

[7] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson, "Averaging weights leads to wider optima and better generalization," in *UAI*, 2018. 1, 3

[8] 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 *ICLR*, 2023. 1, 2, 3, 4

[9] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al., "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *ICML*. PMLR, 2022, pp. 23965–23998. 1

[10] Guoliang Kang, Lu Jiang, Yunchao Wei, Yi Yang, and Alexander G Hauptmann, "Contrastive adaptation network for single-and multi-source domain adaptation," *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[11] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen, "Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation," in *ICCV*, 2022, pp. 7181–7190. 2

[12] Anran Zhang, Yandan Yang, Jun Xu, Xianbin Cao, Xiantong Zhen, and Ling Shao, "Latent domain generation for unsupervised domain adaptation object counting," *IEEE Transactions on Multimedia*, 2022. 2

[13] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*. pmlr, 2015, pp. 448–456. 2, 3, 4

[14] Mario Döbler, Robert A Marsden, and Bin Yang, "Robust mean teacher for continual and gradual test-time adaptation," in *CVPR*, 2023, pp. 7704–7714. 2, 3, 4, 5

[15] Y. Yuan, B. Xu, L. Hou, F. Sun, H. Shen, and X. Cheng, "Tea: Test-time energy adaptation," in *CVPR*, Los Alamitos, CA, USA, jun 2024, pp. 23901–23911, IEEE Computer Society. 2, 3, 4, 5

[16] Maksym Andriushchenko and Nicolas Flammarion, "Towards understanding sharpness-aware minimization," in *ICML*. PMLR, 2022, pp. 639–668. 3

[17] Imad Eddine Marouf, Subhankar Roy, Enzo Tartaglione, and Stéphane Lathuilière, "Weighted ensemble models are strong continual learners," 2024. 3

[18] Gilad Cohen and Raja Giryes, " Simple Post-Training Robustness using Test Time Augmentations and Random Forest ," in *WACV*, Los Alamitos, CA, USA, Jan. 2024, pp. 3984–3994, IEEE Computer Society. 3

[19] Longhui Yuan, Binhui Xie, and Shuang Li, "Robust test-time adaptation in dynamic scenarios," in *CVPR*, 2023, pp. 15922–15932. 3, 4

[20] Yongyi Su, Xun Xu, and Kui Jia, "Towards real-world test-time adaptation: Tri-net self-training with balanced normalization," *AAAI*, vol. 38, no. 13, pp. 15126–15135, Mar. 2024. 3, 4

[21] Dan Hendrycks and Thomas Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *ICLR*, 2019. 3

[22] Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir, "3d common corruptions and data augmentation," in *CVPR*, 2022, pp. 18963–18974. 3

[23] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein, "Robustbench: a standardized adversarial robustness benchmark," in *NeurIPS*, 2021. 3

[24] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017, pp. 1492–1500. 3, 5

[25] Sergey Zagoruyko and Nikos Komodakis, "Wide residual networks," in *BMVC*, 2016. 5