This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

AcTTA: Rethinking Test-Time Adaptation via Dynamic Activation

Hyeongyu Kim[1*] Geonhui Han[1*] Dosik Hwang[1,2†]
[1] Yonsei University [2] Korea Institute of Science and Technology
{lion4309, hgh6945, dosik.hwang}@yonsei.ac.kr
https://hyeongyu-kim.github.io/actta/

Abstract

Test-time adaptation (TTA) aims to mitigate performance degradation under distribution shifts by updating model parameters during inference. Existing approaches have primarily framed adaptation around affine modulation, focusing on recalibrating normalization layers. This perspective, while effective, overlooks another influential component in representation dynamics: ***the activation function****. We revisit this overlooked space and propose* ***AcTTA****, an activation-aware framework that reinterprets conventional activation functions from a learnable perspective and updates them adaptively at test time. AcTTA reformulates conventional activation functions (e.g., ReLU, GELU) into parameterized forms that shift their response threshold and modulate gradient sensitivity, enabling the network to adjust activation behavior under domain shifts. This functional reparameterization enables continuous adjustment of activation behavior without modifying network weights or requiring source data. Despite its simplicity, AcTTA achieves robust and stable adaptation across diverse corruptions. Across CIFAR10-C, CIFAR100-C, and ImageNet-C, AcTTA consistently surpasses normalization-based TTA methods. Our findings highlight activation adaptation as a compact and effective route toward domain-shift–robust test-time learning, broadening the prevailing affine-centric view of adaptation.*

1. Introduction

Deep neural networks often suffer substantial performance degradation when deployed in environments that differ from their training distribution [9, 10, 13]. To address this, test-time adaptation (TTA) has emerged as a paradigm for adapting models during inference without access to source data [17, 26, 27, 29]. Existing approaches span a wide spectrum, from full or partial fine-tuning of model weights to lightweight, layer-specific updates [11, 12, 30]. Among them, normalization-based strategies have become the most prevalent, typically adapting models by updating affine parameters and recalibrating running statistics of normalization layers [15, 23, 27]. These methods have established normalization adjustment as the dominant mechanism for TTA, under the assumption that aligning feature statistics is the most effective way to mitigate domain shift.

However, this assumption leaves a fundamental and highly influential component almost untouched: ***the activation function***. The nonlinearity introduced by activations fundamentally shapes the geometry of feature spaces and governs how representations respond to input variations [1, 6, 25]. Despite this central role, activation functions have been treated as fixed nonlinear mappings during adaptation, ignoring their potential as adaptable mechanisms for domain alignment.

Interestingly, outside the TTA context, a growing body of work has explored learnable or modulated activation functions, ranging from parametric forms to gated variants, which enable richer and more adaptive representation dynamics [2, 7, 18, 21]. These studies consistently show that even subtle modifications in activation behavior can lead to improvements in performance and training stability. This line of work suggests that activation functions inherently possess a learnable degree of variability that can modulate how features respond to input shifts. Yet, this flexibility has remained largely unexplored in the TTA setting, where the role of activation adaptability has been largely overlooked.

Activation functions serve as the nonlinear core of neural networks, shaping how features transform and how gradients propagate. In the context of test-time adaptation, this nonlinearity plays a critical yet underexplored role: it determines how sensitively the model responds to shifted feature distributions. If the activation itself could adapt while adjusting its response center or modulating its gradient flow, the network could internally re-balance without altering its weights. Such adaptability would enable more fine-grained and localized correction of feature drift, complementing existing TTA strategies that typically rely on external normal-

*Equal contribution
†Corresponding author

ization or weight updates. This insight motivates rethinking the activation function as an active participant in adaptation rather than a fixed component.

To explore this idea, we introduce a simple activation-aware TTA framework, termed **AcTTA**, which modulates the parameters shaping the activation shift and gradient flow. By centering adaptation on these activation dynamics, AcTTA allows the model to refine its nonlinear responses to distribution shifts while keeping the overall architecture largely unchanged.

Our contributions can be summarized as follows:

- We introduce **AcTTA**, a new perspective on TTA that leverages adaptive activation modulation as an effective and compact way to handle distribution shifts.
- We design a **modular and easily pluggable activation adaptation module** that enables activation behavior to be dynamically adjusted during inference, without source data or architectural modifications.
- Extensive experiments on CIFAR10-C, CIFAR100-C, and ImageNet-C demonstrate consistent improvements in performance and stability.

2. Related Work

2.1. Test-Time Adaptation

Recent advances in TTA have explored diverse axes of adaptation beyond simple normalization updates. One major line of work builds on entropy minimization, encouraging confident and consistent predictions by directly reducing output uncertainty on unlabeled target samples [23, 27, 29]. Other approaches emphasize consistency regularization or self-ensembling, maintaining temporal or augmentation-level stability to improve adaptation robustness [16, 23, 28]. A complementary direction, often referred to as test-time training, leverages auxiliary self-supervised objectives to refine representations during inference [26]. Together, these studies demonstrate that even lightweight, on-the-fly optimization can substantially mitigate domain shift without source data [12].

Despite this growing diversity, most approaches concentrate adaptation on normalization layers or feature statistics, implicitly assuming that aligning affine parameters is the primary means to handle domain shift. This prevailing focus leaves other influential components such as the activation dynamics that govern nonlinear transformations largely underexplored, motivating our investigation into activation-level adaptation.

2.2. Dynamic Activation Functions

Early studies on activation functions primarily sought fixed nonlinearities that improved optimization stability and representational power. These efforts evolved from early functions such as sigmoid and tanh [14] to ReLU [5, 22] and its modern variants such as Swish [25], ELU [4], and GELU [8], which introduce smoother or self-gated nonlinearities that enhance gradient flow and learning stability. Within this paradigm, the activation function was regarded as a static architectural choice, serving as an essential yet immutable component of the network.

In contrast, more recent research has explored dynamic activation functions that endow networks with functional flexibility [3]. Parametric forms such as PReLU [7] introduced trainable slopes to extend rectifiers, while piecewise-linear and rational formulations like APLU [2] allowed networks to learn customized nonlinear mappings from data. PAU [21] represents activations as learnable rational functions, jointly optimizing numerator and denominator coefficients to approximate a wide range of nonlinear behaviors in an end-to-end manner. ACON [18] introduces a gating-based mechanism that dynamically controls whether and how strongly neurons are activated, enabling input-dependent modulation of nonlinearity. Together, these studies highlight that nonlinear transformations need not remain fixed; they can dynamically adjust and reshape, continuously influencing how representations evolve across domains and conditions.

3. AcTTA

3.1. Dynamic Activation Reparameterization

Normalization layers such as Batch Normalization (BN) aim to stabilize feature distributions by providing approximately zero-centered and standardized inputs to subsequent activations. However, under distribution shift, the source statistics used by BN may no longer align with target features, producing biased representations that remain even after normalization. When such biased features are processed by zero-centered activations like ReLU, Swish, or GELU, informative responses can be suppressed below the activation boundary, leading to information loss and vanishing gradients [20]. We identify this zero-centered rigidity as a key factor limiting adaptation, particularly in TTA settings where feature statistics deviate from the source domain.

Recent works such as ACON [18] have relaxed this rigidity by introducing learnable gating functions that preserve nonzero gradients in the negative region. While such designs mitigate the dead-gradient problem, they still assume a fixed zero-centered boundary and thus cannot effectively handle biased or shifted feature distributions. In test-time adaptation, where the model must adjust to unseen domains without altering its core parameters, this limitation becomes critical. We therefore seek a more general activation formulation that (1) adapts its gradient behavior to sustain learning flow, (2) shifts its activation boundary to align with new feature centers, and (3) remains compatible with the source-trained representation, allowing adaptation to emerge naturally from within the activation function itself.

3.2. AcTTA: Activation Modulation for TTA

Modern activation functions are locally linear mappings whose slope varies smoothly with the input. For instance, Swish and GELU can be approximated as $\phi(x) = x \cdot \sigma(\beta x)$, where the sigmoid gate $\sigma(\cdot)$ controls how much of the input passes through. The derivative,

$$\phi'(x) = \sigma(\beta x) + \beta x\, \sigma(\beta x)\,(1 - \sigma(\beta x)),$$

acts as an *input-dependent slope* that transitions continuously between low and high values. This slope governs gradient propagation, but because it is fixed and zero-centered, it can lead to gradient imbalance or biased updates when feature distributions shift.

To enable adaptive behavior, we explicitly expose this slope as a learnable function:

$$\lambda(x) = \lambda_{\text{neg}} + (\lambda_{\text{pos}} - \lambda_{\text{neg}})\, \sigma(\beta x),$$

where λ_{neg} and λ_{pos} represent the asymptotic slopes in the negative and positive regions. This formulation generalizes the derivative $\phi'(x)$, allowing the gradient scale to adapt asymmetrically across feature dimensions.

However, slope adaptation alone cannot correct feature bias. Under distribution shift, activations often become misaligned because the zero-centered boundary no longer coincides with the shifted feature mean. To address this, we introduce a learnable center parameter c that repositions the activation boundary, allowing the activation to re-center itself dynamically according to the target-domain statistics.

Combining these mechanisms yields the final activation:

$$g(x) = \underbrace{\phi(x - c)}_{\text{shifted base}} + \underbrace{[\lambda_{\text{neg}} + (\lambda_{\text{pos}} - \lambda_{\text{neg}})\sigma(\beta(x - c))]\,(x - c)}_{\text{adaptive slope modulation}} \tag{1}$$

Here, $\phi(\cdot)$ denotes a differentiable base activation (e.g., ReLU or GELU). When initialized with $\lambda_{\text{neg}} = \lambda_{\text{pos}} = 0$ and $c = 0$, $g(x)$ exactly recovers the original $\phi(x)$, ensuring compatibility with a pre-trained model. See Supplementary for detailed derivation.

By jointly learning λ_{neg}, λ_{pos}, and c during adaptation, the activation can locally adjust both its gradient flow and its activation center. This provides a more expressive and stable adaptation mechanism than conventional normalization layers, which only adjust global affine parameters. In effect, AcTTA allows the model to self-correct internal biases and preserve gradient dynamics in a domain-adaptive manner, without modifying network weights or accessing source data.

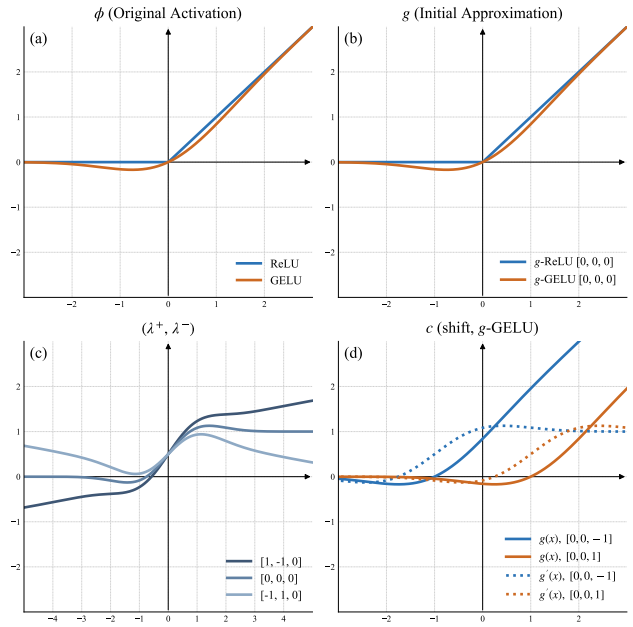

Figure 1. Visualization of the adaptive activation function. (a) The base activation functions $\phi(x)$. (b) The AcTTA activation output $g(x)$; each parameter set $[\cdot, \cdot, \cdot]$ denotes $[\lambda^+, \lambda^-, c]$. (c) The first derivative $g'(x)$ with varying (λ^+, λ^-) (d) The effect of the shift parameter c, which avoids strict zero-centering during adaptation.

3.3. Adaptation Objective

Most recent TTA studies emphasize *how* the model should be updated, designing complex objectives such as consistency regularization, confidence-based selection, or entropy-weighted filtering to achieve stable adaptation. Rather than being tied to a specific update rule, our formulation is motivated by making the activation itself learnable, allowing adaptation to emerge naturally regardless of the employed objective. By simply assigning the learnable target to the activation parameters, the model can self-adjust to distributional shifts without introducing any additional optimization heuristics or auxiliary losses. This minimal yet effective objective highlights the modular nature of our approach: it can be seamlessly integrated with diverse adaptation strategies while maintaining simplicity and interpretability.

4. Experiment

4.1. Implementation Details

We revisit representative TTA methods such as TENT [27], EATA [23], SAR [24], DeYO [15], ROID [19], and CMF [16] which mainly adapt normalization parameters during inference. For EATA, to comply with the fully source-free assumption, we exclude the Fisher regularization and adopt only the sample-selection ($\rightarrow$ ETA).

Our framework AcTTA is modular and objective-agnostic. By making activation parameters learnable alongside affine parameters, it can integrate with existing adaptation objectives (e.g., entropy, consistency, or regularization) without altering their optimization schemes. We denote such variants as $\text{AcTTA}_{\text{TENT}}$, $\text{AcTTA}_{\text{ETA}}$, etc., where the subscript indicates the baseline objective.

Datasets. Experiments are conducted on CIFAR10-C, CIFAR100-C, and ImageNet-C, each with 15 corruption types, with the highest severity level, which is 5. Backbones include WRN-28, ResNeXt, WRN-40, ResNet50 (BN/GN), and ViT-B/16, pretrained on clean training data and adapted online without source access. All methods share identical optimization and hyperparameter settings for fair comparison. All results are averaged over three random seeds. Detailed experimental settings, along with additional results on ImageNet-A, ImageNet-R, and PACS, are provided in the supplementary material.

4.2. Stable Optimization with Larger Learning Rate

Practical TTA, especially in online scenarios, requires the model to adapt immediately as target samples arrive, making fast and responsive updates essential. Although using a large learning rate is desirable for such rapid adaptation, most existing TTA methods easily become unstable or even fail because their gradients vanish in the negative activation region under distribution shift. In contrast, AcTTA preserves nonzero gradients across both positive and negative regions through its adaptive slope design, enabling stable and fast optimization even with aggressive updates. This robustness allows us to safely employ a much larger learning rate; in practice, we use roughly 10× larger than those used in existing TTA methods.

	TENT				$\text{AcTTA}_{\text{TENT}}$			
LR / BS	4	16	64	128	4	16	64	128
1e-2	98.51	94.41	73.28	51.57	55.90	39.03	**34.94**	**34.56**
5e-3	97.61	81.49	43.47	38.46	**55.27**	**38.87**	35.45	35.31
1e-3	87.80	**40.10**	**35.62**	**35.25**	55.66	40.39	37.74	37.61
1e-4	**57.37**	41.54	38.93	38.61	-	-	-	-

Table 1. Sensitivity comparison to learning rate and batch size for TENT and $\text{AcTTA}_{\text{TENT}}$ on CIFAR100-C using WRN-40. Sky blue denotes the best performance.

4.3. Trainable Parameter Ablation

Table 2 analyzes which components of the proposed activation module are most beneficial to update at test time. Starting from the Tent [27] baseline that adapts only the normalization parameters (γ, β) with Entropy Minimization (EM), we progressively introduce the learnable activation parameters of AcTTA: the activation center c and the asymmetric gradient scales $(\lambda_{\text{pos}}, \lambda_{\text{neg}})$. On CNNs with BatchNorm (WRN-28), an interesting pattern emerges. Contrary to the common TTA practice of updating (γ, β), the best performance is obtained when we *freeze* the BN affine parameters and adapt only the activation parameters $(\lambda_{\text{pos}}, \lambda_{\text{neg}}, c)$ (AcTTA*). Updating (γ, β) together with the activation parameters slightly degrades accuracy, suggesting that BN and AcTTA play overlapping roles on convolutional backbones. Since BN already relies on running statistics that are perturbed under shift, further modifying (γ, β) can amplify distributional noise and overcorrect feature scales. In contrast, keeping BN fixed while allowing the activation to adjust its center and asymmetric slopes provides a more localized and stable way to compensate residual bias and preserve gradient flow, which is particularly beneficial under our large learning rate setting.

Method	**Trainable params**	**WRN-28**	**ViT-B/16**
No Adaptation	–	43.52	62.10
Tent	γ, β	18.51	53.85
AcTTA	γ, β, c	18.21	53.89
AcTTA	$\gamma, \beta, \lambda_{\text{pos}}$	18.39	53.85
AcTTA	$\gamma, \beta, \lambda_{\text{neg}}$	18.40	52.40
AcTTA	$\gamma, \beta, \lambda_{\text{pos}}, c$	18.14	53.88
AcTTA	$\gamma, \beta, \lambda_{\text{neg}}, c$	18.13	52.44
AcTTA	$\gamma, \beta, \lambda_{\text{pos}}, \lambda_{\text{neg}}$	18.29	52.38
AcTTA	$\gamma, \beta, \lambda_{\text{pos}}, \lambda_{\text{neg}}, c$	18.06	**52.37**
AcTTA_*	$\lambda_{\text{pos}}, \lambda_{\text{neg}}, c$	**17.03**	55.30
AcTTA_*	λ_{pos}, c	17.15	60.88
AcTTA_*	λ_{neg}, c	17.21	55.58
AcTTA_*	c	17.50	61.56

Table 2. Ablation on trainable parameters of AcTTA at test time (λ_{neg}, λ_{pos}, and c) within the Tent [27] framework. Reported is the error rate on CIFAR10-C (WRN-28) and ImageNet-C (ViT-B/16). AcTTA_* denotes the variant using a learning rate scaled by $\times 10$ (as defined in Table 1). Batch size is 128.

For ViT-B/16 with LayerNorm, the behavior is noticeably different. Here, the best configuration is achieved when the LN affine parameters (γ, β) are adapted jointly with the activation parameters. Unlike BN, LN normalizes each token on a per-sample basis and does not depend on running statistics from the source domain. As a result, adapting (γ, β) in ViTs does not interfere with noisy batch statistics, but instead provides an additional degree of freedom to rescale and re-center features after per-token normalization. This rescaling is complementary to AcTTA's activation-level modulation: (γ, β) adjusts global importance across channels, while $(\lambda_{\text{pos}}, \lambda_{\text{neg}}, c)$ refine the local nonlinear response. The strong gains observed when both are updated indicate that, under LN, normalization and activation adaptation cooperate rather than compete.

Overall, this ablation highlights that the optimal design of trainable components is backbone dependent. For BN-

based CNNs, it is advantageous to depart from the conventional "update only (γ, β)" paradigm and rely primarily on activation parameters for adaptation. For LN-based Transformers, jointly adapting normalization and activation proves more effective. These findings support the modular nature of AcTTA and motivate our default choice of using base AcTTA on CNNs and the full AcTTA configuration on ViTs in subsequent experiments.

A more fine-grained observation concerns the role of the activation-center parameter c. Allowing c to shift already brings noticeable gains, particularly on CNNs with BN, suggesting that residual bias induced by source-domain running statistics can be compensated by adjusting the activation boundary itself. By contrast, Vision Transformers with LN show smaller gains from adapting c, since LN normalizes features on a per-sample basis and is less affected by domain-wise mean shift.

4.4. Comparison on Other Learnable Activations

We further compare our adaptive activation formulation against several representative learnable activation functions, including PReLU [7], (parametric) GELU [8], ACON [18], and PAU [21]. While these methods introduce parametric flexibility to improve expressiveness during training, their structures remain limited to fixed functional forms, which are typically linear or monotonic variants with a small number of learnable coefficients. In contrast, our approach parameterizes the activation center and asymmetric slopes, enabling input-dependent reshaping of the response function. This formulation allows flexible shift and slope modulation, which is particularly beneficial under distributional shifts encountered at test time.

Activation Function	WRN-28	ViT-B/16
Original	18.51	53.85
PReLU / PGELU [7]	29.62	53.87
ACON [18]	18.46	52.92
PAU [21]	34.50	99.96
AcTTA (Ours)	**18.06**	**52.37**

Table 3. Comparison of activation functions under TTA on CIFAR10-C and ImageNet-C under Tent framework. Results are average classification error (%) on WRN-28 and ViT-B/16.

To ensure a fair comparison, we adopt the same adaptation setting (EM) and backbone architectures as in prior sections, replacing the base activation function with each candidate method. As summarized in Table 3, AcTTA delivers strong overall performance across a wide spectrum of datasets and architectural families such as CNN and ViT. Notably, while methods like ACON and PAU provide moderate adaptability through parametric scaling, our formulation yields lower error rates in most settings under corruption shifts. This suggests that test-time adaptation benefits not merely from parameterized slope control but from the joint modulation of the activation center and asymmetric slopes.

In essence, the gain stems from explicitly parameterizing the activation center and asymmetric slopes, allowing the response function to adapt to new domain statistics without overwriting pretrained behavior. This adaptive yet stable mechanism explains why our formulation generalizes better under unseen corruptions than traditional learnable activations with more limited adaptation flexibility.

4.5. Depth-wise Analysis of Learnable Activations

A natural question arises: would making all activation functions learnable always lead to better adaptation? Prior studies often note that domain shifts tend to manifest more strongly in the early feature extraction stages, while deeper layers capture more domain-invariant semantics [12]. Although parameterizing every activation function increases representational flexibility, it also introduces additional computational cost and potentially unstable updates during online adaptation. Striking a balance between adaptability and efficiency is therefore crucial.

Model	∼10%	∼25%	∼50%	∼75%	∼100%
WRN-40	36.17	35.14	34.05	35.13	44.97
ResNet50	67.67	67.02	65.89	63.45	63.47
ViT-B/16	56.29	51.40	52.47	57.81	65.84

Table 4. Effect of learnable depth ratio in AcTTA across different architectures (WRN on CIFAR100-C; ResNet50/ViT on ImageNet-C). Sky-blue cells denote ranges showing performance improvement.

To further investigate this trade-off, we vary how deep into the network activations are made learnable. Our results show that the optimal adaptation depth is architecture-dependent: different architectures exhibit distinct sensitivity profiles, making it difficult to identify the best depth a priori. While deeper adaptation provides more flexibility, it also increases computational cost and the risk of unstable updates, creating a trade-off between effectiveness and reliability. Given these constraints and the practical demands of online TTA, we adopt a mid-depth configuration (50%) as a principled compromise, offering strong and stable performance across architectures without excessive overhead.

4.6. Baseline Comparison: Large Batch Size

As summarized in Table 5, AcTTA provides consistent and substantial improvements over state-of-the-art TTA methods across a wide spectrum of datasets and architectural families. A key strength of our framework lies in its **high modularity**: activation adaptation is introduced as a plug-in and objective-agnostic component, allowing seamless integration with diverse baseline strategies. Despite the

Method	CIFAR10-C		CIFAR100-C		ImageNet-C		
	WRN28	ResNeXt	WRN40	ResNeXt	ResNet50 (BN)	ResNet50 (GN)	ViT-B/16 (LN)
Source	43.52	17.98	46.75	46.45	82.03	69.52	62.10
BN	20.61	13.31	39.51	35.61	68.10	69.52	62.10
TENT [27]	18.51 (±0.01)	10.28 (±0.01)	35.25 (±0.01)	31.25 (±0.05)	66.50 (±0.05)	69.60 (±0.13)	53.85 (±0.18)
AcTTA$_{\text{TENT}}$	17.03 (±0.05)	9.53 (±0.03)	33.81 (±0.09)	30.54 (±0.01)	64.95 (±0.02)	66.84 (±0.02)	51.79 (±0.02)
ETA [23]	18.07 (±0.38)	9.74 (±0.05)	36.33 (±0.01)	31.07 (±0.02)	62.84 (±0.02)	68.34 (±0.03)	49.84 (±0.49)
AcTTA$_{\text{ETA}}$	16.74 (±0.23)	9.46 (±0.03)	34.90 (±0.28)	31.00 (±0.10)	61.74 (±0.02)	61.37 (±0.05)	48.90 (±0.30)
SAR [24]	20.55 (±0.06)	12.97 (±0.00)	37.13 (±0.04)	33.03 (±0.02)	66.24 (±0.04)	69.37 (±0.07)	53.90 (±0.09)
AcTTA$_{\text{SAR}}$	20.54 (±0.01)	12.95 (±0.01)	35.90 (±0.11)	32.38 (±0.02)	64.43 (±0.02)	66.62 (±0.02)	54.17 (±0.19)
ROID [19]	17.52 (±0.06)	9.62 (±0.05)	35.38 (±0.06)	30.12 (±0.13)	59.17 (±0.09)	64.39 (±0.05)	52.91 (±0.11)
AcTTA$_{\text{ROID}}$	16.43 (±0.08)	9.43 (±0.01)	33.59 (±0.01)	**29.49** (±0.26)	59.74 (±0.08)	55.87 (±0.02)	49.89 (±0.26)
CMF [16]	17.36 (±0.08)	9.67 (±0.05)	35.27 (±0.07)	30.05 (±0.10)	**59.06** (±0.05)	64.30 (±0.09)	50.85 (±0.49)
AcTTA$_{\text{CMF}}$	**16.39** (±0.44)	**9.41** (±0.01)	33.57 (±0.12)	29.62 (±0.20)	59.95 (±0.02)	**55.46** (±0.31)	50.44 (±0.02)
DeYO [15]	18.31 (±0.07)	9.73 (±0.08)	35.17 (±0.04)	30.28 (±0.14)	62.34 (±0.05)	69.02 (±0.13)	**47.73** (±0.30)
AcTTA$_{\text{DeYO}}$	17.11 (±0.09)	9.56 (±0.01)	**33.55** (±0.53)	30.04 (±0.72)	61.79 (±0.04)	63.36 (±0.24)	49.64 (±0.19)

Table 5. Comparison with state-of-the-art TTA methods on **CIFAR10-C**, **CIFAR100-C**, and **ImageNet-C** (severity level 5). Bold numbers indicate the best results in each column.

heterogeneous nature of these baselines, each relying on distinct optimization dynamics and statistical assumptions, AcTTA enhances their performance without requiring any modification to their core objectives or update rules. This confirms that activation-level modulation acts as a broadly compatible mechanism that complements, rather than competes with, existing adaptation pipelines.

Across CIFAR10-C, CIFAR100-C, and ImageNet-C, AcTTA reduces error rates in most evaluated architectures, including WRN, ResNeXt, ResNet50 with both BN and GN, and ViT-B/16 with LayerNorm. The breadth of these improvements highlights the generality of our formulation. By endowing the network with adaptive activation mechanisms, AcTTA enables a more flexible functional response that can absorb distribution shifts independently of the underlying normalization scheme. Although the magnitude of improvement varies across baselines, AcTTA generally yields positive gains in most settings. This reinforces that activation adaptation introduces a complementary capability that operates orthogonally to normalization-based updates, providing a robust and architecture-agnostic avenue for enhancing test-time performance.

Notably, the performance lift is most pronounced on ImageNet-C, where distribution shifts are more complex and high-capacity models tend to amplify instability during online adaptation. The fact that AcTTA maintains robustness in these challenging settings indicates that its modulation of activation functions yields a more structurally reliable adaptation path than relying solely on normalization parameters. Together, these findings validate the effectiveness and versatility of activation-aware test-time adaptation, establishing AcTTA as a practical and scalable enhancement to a wide range of TTA methodologies.

4.7. Baseline Comparison: Small Batch Size

Method	CIFAR10-C	CIFAR100-C	ImageNet-C
	WRN28	WRN40	ViT-B/16
Source	43.52	46.75	62.10
TENT [27]	30.36	57.35	89.92
AcTTA$_{\text{TENT}}$	29.56	**55.03**	**62.07**
ETA [23]	30.09	57.49	98.37
AcTTA$_{\text{ETA}}$	29.11	57.04	72.70
SAR [24]	31.49	57.20	71.81
AcTTA$_{\text{SAR}}$	31.03	56.93	82.11
ROID [19]	30.87	58.75	99.80
AcTTA$_{\text{ROID}}$	29.62	56.38	86.37
CMF [16]	30.19	58.27	99.79
AcTTA$_{\text{CMF}}$	**28.64**	55.72	87.08
DeYO [15]	30.23	56.89	99.45
AcTTA$_{\text{DeYO}}$	29.25	55.14	79.49

Table 6. Comparison with state-of-the-art TTA methods on small batch size regime (batch size =4). Reported are average classification errors (%), averaged over 3 runs. Bests are in bold.

As shown in Table 6, AcTTA remains effective even under the small-batch regime (batch size = 4), where test-time adaptation is typically more unstable due to noisy gradient estimates and unreliable normalization statistics. Despite these challenges, our activation-based adaptation consistently outperforms baseline methods across CIFAR10-C, CIFAR100-C, and ImageNet-C. This robustness highlights that activation modulation remains an effective adaptation mechanism even when statistical signals are severely limited, a scenario where normalization-based approaches often degrade sharply.

It is important to note that all baseline methods are evaluated using their optimal learning rates, as identified in Ta-

		Time ⟶															
	Method	*gaussian*	*shot*	*impulse*	*defocus*	*glass*	*motion*	*zoom*	*snow*	*frost*	*fog*	*brightness*	*contrast*	*elastic*	*pixelate*	*jpeg*	Mean
Target	No Adaptation	65.69	60.05	59.10	32.04	50.96	33.57	32.35	41.39	45.17	51.38	31.64	55.51	40.24	59.70	42.43	46.75
	BN Adapt	45.28	44.17	47.58	32.32	46.21	32.97	33.00	39.05	38.53	45.46	30.39	36.70	40.84	37.19	43.99	39.58
	TENT-continual	40.31	**37.29**	41.49	33.55	43.52	35.50	32.89	39.20	38.45	42.91	36.26	41.63	47.15	45.67	57.28	40.87
	CoTTA	43.75	41.86	43.72	32.81	43.04	33.15	33.06	38.10	37.59	45.54	29.90	37.76	39.27	33.86	39.91	38.22
	AcTTA$_{\text{TENT}}$	**39.42**	37.40	**41.20**	**30.29**	**42.96**	**30.52**	**30.31**	**34.95**	**33.58**	**37.21**	**28.39**	**29.57**	**37.23**	**31.29**	**39.05**	**34.89**
Source	BN Adapt	27.69	**27.37**	**27.49**	27.84	27.58	27.69	27.52	27.63	27.63	27.55	27.56	27.46	28.03	27.49	**27.47**	27.60
	TENT-continual	**26.65**	27.88	30.22	29.28	30.53	30.27	29.82	30.75	31.24	33.67	33.73	35.17	38.93	41.08	48.40	33.17
	CoTTA	34.31	35.06	36.02	34.59	34.73	35.12	34.23	34.09	34.05	33.39	33.49	34.65	33.65	34.34	34.05	34.38
	AcTTA$_{\text{TENT}}$	28.40	29.08	28.15	**25.56**	**25.97**	**25.12**	**24.78**	**25.67**	**26.16**	**26.78**	**25.04**	**25.32**	**25.11**	**25.63**	28.32	**26.34**

Table 7. Error rate comparison (%) on **CIFAR100-C** corruptions over time under the continual test-time adaptation setting. Bold numbers indicate the best results in each column.

ble 1 and validated for fairness. Nevertheless, AcTTA provides generally positive gains across diverse architectures and objectives. This result reinforces that the advantage of our approach does not arise from hyperparameter tuning or favorable optimization settings, but from the intrinsic stability and flexibility of activation-level adaptation. The ability to deliver reliable improvements under both large- and small-batch conditions further demonstrates the practical applicability of AcTTA in real-world TTA scenarios, where batch sizes are often constrained by latency or memory limits.

4.8. Continual Forgetting

While continuous test-time adaptation is often regarded as a key requirement for deployment in dynamic real-world environments, we argue that an equally fundamental yet under-explored challenge lies in *continual forgetting*. Existing TTA works predominantly focus on achieving smooth adaptation as the target distribution evolves, and typically measure success only in terms of target-domain accuracy. However, repeated and accumulated updates on non-stationary target streams can gradually overwrite the knowledge acquired from the source domain, causing degradation in performance. This issue is particularly severe in online and long-horizon settings, where access to source data is prohibited and no explicit mechanism exists to recover from overfitting to transient target distributions.

From this perspective, we advocate evaluating TTA methods not only by how well they adapt to continuous domain shifts, but also by how effectively they **preserve source-domain competence** during adaptation. Our activation-based framework directly addresses this trade-off: AcTTA performs strong adaptation on continuously evolving corruptions while exhibiting substantially reduced forgetting of the pretrained behavior. By enabling flexible yet stable modulation of activation responses, our method maintains robust performance on both source and target domains over extended adaptation horizons. We therefore posit that *continual forgetting* should be treated as a primary diagnostic for practical TTA, and that activation-aware adaptation offers a principled avenue toward mitigating this phenomenon.

A deeper examination reveals that continual forgetting is not merely a side effect of unstable optimization, but a structural vulnerability of existing TTA pipelines. Methods that update normalization statistics or affine parameters at every step can inadvertently bias the representation toward the current mini-batch, especially under severe corruptions or rapidly changing target streams. Since these updates accumulate without any mechanism for reversibility, the model progressively drifts away from the source-trained manifold, resulting in catastrophic degradation when the corruption pattern fluctuates or returns to a distribution closer to the source. In contrast, AcTTA limits adaptation to the activation space, enabling the model to adjust its nonlinear response characteristics without perturbing the core representational geometry encoded by the pretrained weights. This design inherently constrains how much the model can deviate from its original semantics, mitigating long-term drift while still enabling meaningful adaptation. Consequently, AcTTA achieves a more stable equilibrium between reactivity and retention, making it well-suited for long-horizon, real-world deployment scenarios where distribution shifts evolve continuously and unpredictably.

4.9. Dynamic Activation Behavior Analysis

To better understand the effect of AcTTA during adaptation, we analyze how the activation functions themselves evolve after test-time learning. Figure 2 visualizes several representative activations extracted from the adapted model. We observe that the learned activation boundaries shift noticeably compared to the source-trained counterparts, confirm-

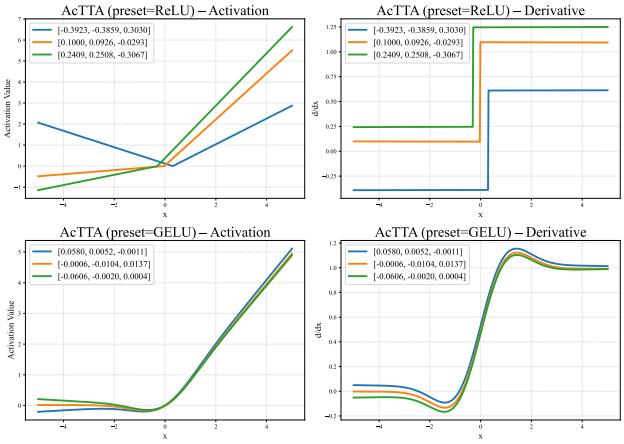

Figure 2. Visualizations of adapted activations during AcTTA. AcTTA-ReLU results are obtained from WRN-28 on CIFAR10-C after completing adaptation on the Gaussian noise corruption. AcTTA-GELU results are from ViT-B/16 on ImageNet-C at the same adaptation endpoint. The learned activation exhibits flexible, channel-specific shaping behavior, adapting its form according to the input distribution. The three legend values denote the **adapted** parameter values of AcTTA, $[\lambda^+, \lambda^-, c]$, corresponding to positive/negative slope modulation and shift.

ing that AcTTA effectively recenters the nonlinear response to match the target-domain feature statistics. At the same time, the asymmetric slope parameters adjust to preserve nonzero gradients in regions where conventional activations would typically saturate. These structural changes are not manually imposed but emerge naturally through the optimization of $(\lambda_{\text{pos}}, \lambda_{\text{neg}}, c)$ during online adaptation.

Interestingly, the adapted activations differ across layers: early layers tend to exhibit larger center shifts, compensating for distributional biases introduced by corrupted inputs, whereas deeper layers mainly refine gradient scales to stabilize optimization. This layer-specific behavior suggests that AcTTA provides a flexible mechanism that automatically allocates adaptation capacity where it is most needed, without modifying the network weights or relying on source data. Overall, these observations offer concrete evidence that activation-level reparameterization forms a meaningful internal adjustment pathway, enabling the network to realign its representation space and maintain stable gradient flows under challenging target-domain shifts.

4.10. Resource-Performance Trade-off

Table 8 presents the resource–performance trade-off of our method compared to existing TTA approaches. Although AcTTA introduces additional learnable activation parameters and maintains gradient flow through them, resulting in a moderate increase in memory usage, it remains substantially more efficient than recent complex TTA algorithms such as CMF and ROID, both of which incur significantly higher VRAM and computational overhead. Despite its lightweight design, AcTTA achieves the lowest error rate among all compared methods, demonstrating that activation-level adaptation offers a highly favorable balance between computational cost and adaptation effectiveness. This confirms that meaningful improvements in TTA robustness do not require heavy architectural modifications or memory-intensive consistency pipelines; instead, fine-grained modulation of activation dynamics provides a compact yet powerful mechanism for reliable adaptation.

Method	VRAM (MiB)	Time (ms)	Error (%)
TENT [27]	4013	112.83	18.51
ROID [19]	9785	208.01	17.52
CMF [16]	10439	218.21	17.36
AcTTA$_{\text{TENT}}$ (Ours)	6249	164.51	**17.03**

Table 8. Computational efficiency evaluation. Experiments were conducted on CIFAR10-C using WRN-28 with a batch size of 128. Reported VRAM corresponds to the peak memory usage. AcTTA$_{\text{TENT}}$ incurs higher memory and runtime than TENT, but attains the lowest error, outperforming ROID and CMF while remaining substantially more efficient than both.

5. Conclusion

We presented AcTTA, a new activation-based perspective on TTA. By parameterizing the activation center and asymmetric slopes, AcTTA mitigates feature bias and preserves gradient flow under distribution shift, while remaining compatible with diverse adaptation objectives.

Experiments on various benchmark datasets show that AcTTA delivers strong overall performance across CNNs and Transformers. Depth-wise analyses show that activation-level adaptation offers favorable trade-offs between robustness and efficiency, while remaining effective under challenging conditions such as small batches, aggressive learning rates, and long-horizon continual adaptation.

AcTTA achieves a strong memory–performance trade-off compared with optimization-heavy adaptation methods, making it a practical alternative to normalization-centric approaches. Beyond its empirical effectiveness, AcTTA offers a new perspective on test-time adaptation by using activation-function modulation as a primary mechanism for online adaptation. At the same time, AcTTA has a clear limitation: its effectiveness depends on design choices such as backbone architecture, activation type, and adaptation depth, and these factors are not yet determined in a fully principled manner. This limits the universality of the method and highlights the need for more systematic selection strategies in future work.

Acknowledgements. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-16070382, RS-2025-02215070, RS-2025-02217919), Artificial Intelligence Graduate School Program at Yonsei University (RS-2020-II201361), the Korea Institute of Science and Technology (KIST) Institutional Program under Grant 26E0170.

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[2] Forest Agostinelli, Matthew Hoffman, Peter Sadowski, and Pierre Baldi. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830*, 2014.

[3] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. A survey on modern trainable activation functions. *Neural Networks*, 138:14–32, 2021.

[4] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 4(5):11, 2015.

[5] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[6] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *International conference on machine learning*, pages 1319–1327. PMLR, 2013.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[8] D Hendrycks. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

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

[10] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[11] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34: 2427–2440, 2021.

[12] Hyeongyu Kim, GeonHui Han, and Dosik Hwang. Buffer layers for test-time adaptation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[13] 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 *International conference on machine learning*, pages 5637–5664. PMLR, 2021.

[14] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002.

[15] 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. *arXiv preprint arXiv:2403.07366*, 2024.

[16] Jae-Hong Lee and Joon-Hyuk Chang. Continual momentum filtering on parameter space for online test-time adaptation. In *The Twelfth International Conference on Learning Representations*, 2024.

[17] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

[18] Ningning Ma, Xiangyu Zhang, Ming Liu, and Jian Sun. Activate or not: Learning customized activation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8032–8042, 2021.

[19] Robert A Marsden, Mario Döbler, and Bin Yang. Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2555–2565, 2024.

[20] Sina Mohseni, Arash Vahdat, and Jay Yadawa. Shifting transformation learning for out-of-distribution detection. *arXiv preprint arXiv:2106.03899*, 2021.

[21] Alejandro Molina, Patrick Schramowski, and Kristian Kersting. Padé activation units: End-to-end learning of flexible activation functions in deep networks. *arXiv preprint arXiv:1907.06732*, 2019.

[22] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[23] 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.

[24] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

[25] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[26] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020.

[27] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[28] 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*, pages 7201–7211, 2022.

[29] Zixin Wang, Yadan Luo, Liang Zheng, Zhuoxiao Chen, Sen Wang, and Zi Huang. In search of lost online test-time adaptation: A survey. *International Journal of Computer Vision*, 133(3):1106–1139, 2025.

[30] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.