P-DORM: PARAMETER-WISE DYNAMIC OPTIMIZER-STATE RESET FOR TEST-TIME ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time adaptation (TTA) adapts deployed models to unlabeled test streams by optimizing unsupervised objectives on incoming batches. These updates commonly rely on optimizers that accumulate gradient history. We find that this history can become stale after distribution shifts and hinder subsequent adaptation. We therefore propose the **Parameter-wise Dynamic Optimizer-State Reset Mechanism (P-DORM)**, a plug-and-play wrapper that detects stale history by comparing the current gradient with the historical direction stored by the optimizer. We prove that raw directional alignment is not comparable across tensors because its center and fluctuation depend on tensor dimension and directional geometry. P-DORM therefore standardizes each tensor's alignment against its own recent history. When parameter-aware thresholds identify unusually large departures, P-DORM resets the corresponding tensor-specific optimizer states without reverting adapted model parameters or modifying the adaptation objective or network architecture. Extensive experiments across corruption and natural distribution shifts, multiple TTA methods, optimizer families, architectures, and test-stream protocols demonstrate improved adaptation performance and broad plug-and-play compatibility.

1 INTRODUCTION

Deep neural networks perform well when training and test distributions match, but their accuracy can degrade sharply under corruptions or natural distribution shifts caused by changes in environments, sensors, acquisition pipelines, or population composition (Hendrycks & Dietterich, 2019; Koh et al., 2021). Test-time adaptation (TTA) addresses this challenge by updating deployed models online using unlabeled test samples, without accessing source data or modifying source training (Wang et al., 2021; Liang et al., 2025). Most TTA approaches are gradient based: they differentiate an unsupervised adaptation objective on each incoming batch and use the resulting gradients to update selected model parameters (Wang et al., 2021; Tan et al., 2025; Lee et al., 2024). Existing methods promote reliable adaptation through entropy-based objectives, sample selection, drift constraints, selective parameter updates, or model restoration (Wang et al., 2021; Tan et al., 2025; Lee et al., 2024; Press et al., 2023; Yuan et al., 2024; Zhang et al., 2025; Ma et al., 2025; Kim et al., 2025; Huang et al., 2026).

These methods commonly rely on optimizers that accumulate gradient history to stabilize updates and mitigate sampling noise, such as Adam and stochastic gradient descent (SGD) with momentum (Kingma & Ba, 2015; Sutskever et al., 2013). We use SGD to denote this momentum-based variant throughout. Following Sutskever et al. (2013), the SGD update is

$$u^t = \beta_m u^{t-1} + g^t, \qquad \theta^t = \theta^{t-1} - \eta u^t, \tag{1}$$

where g^t is the current gradient, u^t is the momentum buffer, and β_m and η denote the momentum coefficient and learning rate. The momentum buffer is an optimizer state that carries historical directional information across updates, making the optimizer *stateful*. Other widely used optimizers maintain similar history-dependent states, despite differences in their update rules (Kingma & Ba, 2015; PyTorch Contributors, 2026; Chen et al., 2023; Zhuang et al., 2020). However, existing TTA methods primarily regulate objectives, samples, or model parameters, paying little attention to whether the accumulated optimizer history remains compatible with the current test distribution.

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

Recent theory on non-stationary stochastic optimization shows that, although an optimizer's accumulated momentum suppresses gradient noise, it can retain update directions from earlier objectives and increase tracking lag when the objective changes over time. In drift-dominated regimes, momentum may even underperform momentum-free updates (Sahu et al., 2026). Despite relying on a simplified strongly convex and smooth setting, this result highlights the need to actively manage optimizer states in dynamic test environments. The issue is particularly acute in TTA, where test distributions may shift substantially, abruptly, or recurrently, while unsupervised gradients are inherently noisy (Liang et al., 2025). This motivates the hypothesis that resetting stale optimizer states after distribution shifts can alleviate optimization inertia. To examine this hypothesis, Figure 1 compares unmodified optimization (Pure) with optimizer-state resets at known domain boundaries (Boundary-Oracle) across five optimizers. In this experiment, Boundary-Oracle resets the optimizer state at each known transition in an ImageNet-C stream comprising 15 severity-5 corruption domains. Appendix E provides the complete domain and protocol definitions. Boundary-Oracle consistently lowers error, supporting the hypothesis that resetting stale optimizer states can alleviate such inertia.

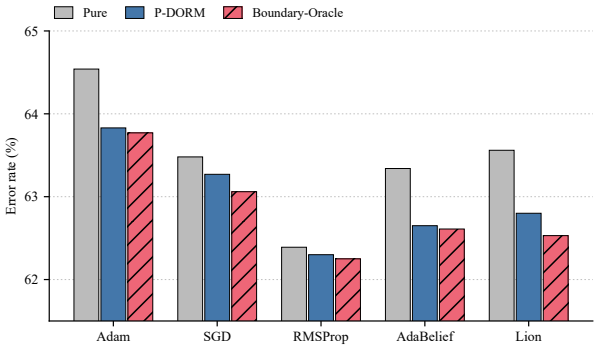


Figure 1: **Optimizer-state reset across five optimizers.** Error rates of TENT (Wang et al., 2021) on ImageNet-C (Hendrycks & Dietterich, 2019) under the continual TTA (CTTA) setting with Pure, P-DORM, and Boundary-Oracle. Pure uses the unmodified optimizer. Boundary-Oracle resets optimizer states at known domain boundaries. Lower is better.

To address stale optimizer memory while preserving adapted model parameters, we introduce a plug-and-play wrapper for stateful optimizers, termed the **Parameter-wise Dynamic Optimizer-State Reset Mechanism (P-DORM)**. To distinguish unusual directional departures from ordinary gradient noise, we compare the current gradient with the optimizer's own pre-update direction and standardize their alignment using tensor-specific online statistics, forming our *Directional Gradient Alignment Criterion*. We further prove that the fluctuation of raw alignment depends on tensor dimension and directional geometry, making raw alignment incomparable across tensors and motivating tensor-specific standardization (Appendix B). To accommodate heterogeneous tensor responses and module structure (Chatterji et al., 2020; Zhang et al., 2022), we combine tensor-specific background thresholds with module-size-dependent sensitivity coefficients, assigning lower coefficients to smaller modules while preserving the network-wide mean. We term this design *Parameter-Aware Threshold Allocation*. When a threshold is exceeded, we reset only the corresponding tensor's optimizer state without reverting model parameters. Figure 1 further shows that P-DORM closely matches the Boundary-Oracle across all five optimizers. P-DORM therefore integrates into TTA without changing the adaptation objective or network architecture.

Our contributions are threefold:

- We identify stale optimizer history as an underexplored failure mode in non-stationary TTA. A Boundary-Oracle study across five optimizer families confirms that resetting optimizer states at distribution boundaries consistently improves adaptation.
- We prove that raw gradient–state alignment is not comparable across tensors because the alignment statistics depend on tensor dimension and directional geometry, and design a tensor-specific online standardization scheme to remove these differences.
- We propose P-DORM, a plug-and-play wrapper that detects directional anomalies with parameter-aware thresholds and resets stale optimizer states without reverting model parameters. Experiments on four datasets, six TTA methods, and five test-stream protocols demonstrate improved adaptation performance and stability.

2 Related Work

2.1 Test-Time Adaptation

Test-time adaptation (TTA) adapts pretrained models to unlabeled test data without source data or retraining (Wang et al., 2021; Liang et al., 2025). Existing methods regulate unsupervised objectives (Wang et al., 2021; Tan et al., 2025; Zhang et al., 2025; Yuan et al., 2024), samples and prototypes (Lee et al., 2024; Huang et al., 2026), drift (Tan et al., 2025), trainable components (Wang et al., 2021; Kim et al., 2025), or stability through sparse activation, teacher averaging, sharpness-aware updates, and model restoration (Ma et al., 2025; Wang et al., 2022; Niu et al., 2023; Press et al., 2023). They rarely examine whether optimizer history remains compatible with evolving test distributions.

2.2 Model Restoration and Optimizer-State Management

Model restoration mitigates continual-TTA drift through stochastic, conditional, or periodic recovery of source parameters (Wang et al., 2022; Niu et al., 2023; Press et al., 2023; Lim et al., 2026). These methods target model parameters rather than stale optimizer history.

Beyond TTA, adaptive restart discards counterproductive momentum, while deep-RL methods reset the entire optimizer at known outer-loop boundaries (O'Donoghue & Candès, 2015; Asadi et al., 2023). Cautious Optimizers mask gradient-conflicting update coordinates rather than clearing history, whereas adaptive-memory optimizers adjust momentum and MGTTA learns a replacement rule (Liang et al., 2026; Topollai & Choromanska, 2026; Deng et al., 2025). P-DORM instead uses tensor-specific standardized statistics to detect incompatible history in unlabeled TTA without boundary annotations. Upon a trigger, it clears only the affected tensor's state, preserves model parameters, and applies the current gradient through the original optimizer update.

3 Method

3.1 Problem Definition

Let f_{θ^0} be a source-pretrained model and $\{X^t\}_{t=1}^{T}$ an unlabeled test stream, where X^t denotes the incoming batch at step t. We use $\mathcal{D}^t$ for the set of admissible test batches in the current domain, so $X^t \in \mathcal{D}^t$. This collection may change over time and is not an underlying probability distribution. Appendix H.3 provides a further clarification. Without source data or target labels, TTA first differentiates an unsupervised adaptation objective $\mathcal{L}_{\mathrm{TTA}}$ on the incoming batch:

$$g^t = \nabla_{\theta^{t-1}} \mathcal{L}_{\mathrm{TTA}}(f_{\theta^{t-1}}; X^t). \tag{2}$$

The optimizer then uses g^t and the pre-update optimizer state to update θ^{t-1} to θ^t, inducing the model transition $f_{\theta^{t-1}} \to f_{\theta^t}$. For clarity, we instantiate this update with SGD as defined in equation (1). The momentum buffer u^{t-1} aggregates preceding gradients and provides the historical update direction. Section 3.2 detects when this buffer becomes incompatible with the current gradient, and Section 3.3 converts the detected departure into parameter-wise thresholding and a tensor-specific reset. Extensions to other optimizers are given in Appendix D.

3.2 Directional Gradient Alignment Criterion

Optimizer history and gradient alignment. P-DORM aims to reset optimizer states that become incompatible with the current optimization direction after distribution shifts. Since domain boundaries are unavailable online, the first challenge is to infer such incompatibility from the adaptation dynamics themselves. The optimizer state aggregates update directions favored by preceding test batches and therefore represents optimization history, whereas the current gradient reflects the objective induced by the present batch (Sutskever et al., 2013; Kingma & Ba, 2015). Comparing their directional discrepancy therefore provides a local signal that the accumulated history may have become incompatible after a distribution shift. Appendix H.1 further discusses the scope of this interpretation.

Let g^t and u^{t-1} denote the current gradient and pre-update momentum buffer associated with the same trainable parameter. We compare these present and historical directions using cosine alignment.

$$C^t = \frac{\langle g^t, u^{t-1} \rangle}{\|g^t\|_2 \, \|u^{t-1}\|_2}, \tag{3}$$

which isolates directional disagreement from gradient magnitude. A smaller C^t indicates a stronger departure from the accumulated direction. We evaluate it before any reset or optimizer update at step t.

Online alignment statistics and anomaly score. To improve detection reliability, the detector must distinguish two sources of directional departure: anomalies induced by distribution shifts and random fluctuations caused by ordinary within-distribution gradient noise. Directly using the raw alignment C^t cannot provide a consistently calibrated signal across different parameter tensors. Appendices B.1–B.3 show that, even under a stationary distribution, tensors can exhibit substantially different normal alignment centers and fluctuation scales because of their dimensions and directional geometry. Consequently, the same absolute decrease may be routine noise for one tensor but an extreme event for another. Because these background distributions differ across tensors, a shared raw threshold yields tensor-dependent false-alarm and miss rates: highly variable tensors become overly sensitive, whereas less variable ones may fail to detect genuine shifts.

To address this problem, P-DORM maintains each tensor's recent alignment level and fluctuation scale. It centers the current alignment relative to that level and normalizes the resulting deviation by the estimated scale, defining the standardized anomaly score

$$Z^t = \frac{E^{t-1} - C^t}{\sqrt{V^{t-1}} + \varepsilon_{\text{var}}}, \tag{4}$$

where E^{t-1} and V^{t-1} track the recent alignment center and variance, and $\varepsilon_{\text{var}} > 0$ ensures numerical stability. Thus, Z^t measures how unusual the current decrease is relative to the normal variation of the tensor, rather than merely measuring the raw cosine magnitude. The statistics are updated jointly by

$$\begin{cases} E^t = \alpha E^{t-1} + (1-\alpha) C^t, \\ V^t = \beta V^{t-1} + (1-\beta)(C^t - E^{t-1})(C^t - E^t), \end{cases} \tag{5}$$

where $\alpha, \beta \in (0,1)$ control the decay of the alignment center and variance, respectively. The first valid alignment initializes $E^{t_0} = C^{t_0}$ and $V^{t_0} = \varepsilon_{\text{var}}$. At each subsequent step, P-DORM computes the anomaly score and reset decision using E^{t-1} and V^{t-1} before updating them with C^t, preventing the current observation from influencing its own score. Appendix B.4 gives the conditions under which this normalization removes tensor-specific location and scale.

The calibration is also supported empirically. Table 1 compares P-DORM using Z^t with an otherwise identical variant using the raw alignment C^t, as defined in Appendix B.5. The standardized score yields lower error in all representative configurations, confirming the importance of tensor-specific calibration.

Table 1: **Alignment C^t vs. Z^t.** ImageNet-C error (%) with ResNet-50 and Adam. Here, $\Delta = \text{Err}(Z^t) - \text{Err}(C^t)$, and negative Δ favors Z^t. Lower is better.

Method / setting	C^t	Z^t	Δ
EATA-C / CTTA	66.79	66.23	**-0.56**
EATA-C / PTTA	67.36	66.87	**-0.49**
DeYO / CTTA	65.11	64.90	**-0.21**
DeYO / PTTA	65.77	65.57	**-0.20**
TENT / CTTA	63.94	63.83	**-0.11**
TENT / PTTA	64.66	64.58	**-0.08**

3.3 Parameter-Wise Reset Thresholding and Optimizer-State Reset

Section 3.2 constructs a calibrated directional anomaly score for identifying optimizer-history incompatibility under distribution shifts. This section converts that score into a conservative reset decision: P-DORM resets an optimizer state only when Z^t significantly exceeds the recent within-distribution range of the score, suppressing noise-induced triggers while retaining responses to substantial distribution shifts.

Algorithm 1 P-DORM update for one parameter tensor with SGD

Require: Gradient g^t, pre-update buffer u^{t-1}, detector state $(E, V, \mathcal{W}, \mu, \sigma, N)$
Require: k_l, (α, β), FIFO capacity W, $W_{\min}$, ε_{var}, $\varepsilon_{\text{norm}}$
if $\|g^t\|_2 \geq \varepsilon_{\text{norm}} \wedge \|u^{t-1}\|_2 \geq \varepsilon_{\text{norm}}$ **then**
  Compute C^t using equation (3)
  if $N > 1$ **then**
    Compute Z^t using equation (4)
    Compute Reset_l^t using equation (8)
    if $\text{Reset}_l^t = 1$ **then**
      $(u^{t-1}, \mathcal{W}, \mu, \sigma) \leftarrow (0, \varnothing, 0, 0)$
    else
      Append Z^t to the FIFO window $\mathcal{W}$ and update (μ, σ)
    end if
  end if
  Initialize (E, V) with $(C^t, \varepsilon_{\text{var}})$ if $N = 0$; otherwise use equation (5)
  $N \leftarrow N + 1$
end if
Apply the original SGD update using g^t

Background statistics and reset criterion. Following statistical anomaly detection based on deviations from an estimated background distribution, P-DORM identifies directional departures relative to tensor-specific online statistics (Chandola et al., 2009; Gupta et al., 2014). For each monitored parameter tensor, P-DORM maintains a First-In, First-Out (FIFO) window $\mathcal{W}^t$ containing at most W valid anomaly scores Z^τ, where $n^t = |\mathcal{W}^t|$ is the number of valid scores currently stored in the window. After a non-triggering step, the current score Z^t is appended. If the capacity is exceeded, the oldest score is removed. From this window, we calculate the mean and sample variance as

$$\begin{cases} \mu^t = \dfrac{1}{n^t} \displaystyle\sum_{\tau \in \mathcal{W}^t} Z^\tau, \\ (\sigma^t)^2 = \dfrac{1}{n^t - 1} \displaystyle\sum_{\tau \in \mathcal{W}^t} (Z^\tau - \mu^t)^2. \end{cases} \tag{6}$$

At $t = 1$, no historical direction is available, so no alignment score is formed and $\mathcal{W}^t$ remains empty. For the first valid score, we set $\mu^t = Z^t$ and $\sigma^t = 0$; the sample variance in equation (6) is evaluated only when $n^t \geq 2$. To avoid unstable thresholds from insufficient history, reset is enabled only after at least $W_{\min}$ valid scores have accumulated. The same condition provides a minimum post-reset recalibration period. At step t, the statistics of the pre-update window define the reset threshold

$$T_{\text{reset}}^t = \mu^{t-1} + k_l \sigma^{t-1}. \tag{7}$$

The generic reset criterion is

$$\text{Reset}^t = \mathbf{1}\left[n^{t-1} \geq W_{\min} \ \wedge \ Z^t > T_{\text{reset}}^t\right]. \tag{8}$$

Here $\mathbf{1}[\cdot]$ is the indicator function and $\wedge$ denotes logical conjunction. A reset thus requires sufficient background observations and a departure beyond the tensor's normal variation. The tensor-specific coefficient k_l controls the conservativeness of this decision, and Section 4.3 evaluates sensitivity to the default choice.

Parameter-aware threshold allocation. Deep-network modules are functionally heterogeneous (Chatterji et al., 2020; Zhang et al., 2022). We further hypothesize that smaller modules respond more sensitively to distribution shifts and can be reset with lower intervention cost. P-DORM therefore assigns them lower reset coefficients, while larger modules receive more conservative thresholds. We evaluate this size prior empirically in our ablations. To implement the allocation, let $\{\theta_l\}_{l=1}^L$ denote the L trainable parameter tensors, each with a separate detector. P-DORM derives the coefficient k_l in equation (7) from module size. Let $m(l)$ be the module containing θ_l, n_m the largest trainable-tensor size in module m, and $M \geq 2$ the number of trainable modules. Sorting $\{\log n_m\}_{m=1}^M$ in ascending order, with tied sizes assigned their average rank, gives

$$w_m = \frac{\text{rank}_{\text{avg}}(\log n_m) - 1}{M - 1}, \tag{9}$$

where $\mathrm{rank}_{\mathrm{avg}}$ is the one-based ascending rank, with tied modules assigned their mean rank, and $w_m \in [0, 1]$ is the resulting normalized module-size rank. Tensor l receives the coefficient assigned to module $m(l)$:

$$k_l = \mathrm{clip}\big(k_0 + \lambda(2w_{m(l)} - 1) + \delta_k^\star, k_{\min}, k_{\max}\big)\,, \tag{10}$$

Here k_0 is the shared baseline, λ controls size dependence, and $\mathrm{clip}(x, a, b) = \min(\max(x, a), b)$ truncates the coefficient to $[k_{\min}, k_{\max}]$. We use symmetric bounds $k_{\min} = k_0 - 0.5$ and $k_{\max} = k_0 + 0.5$, limiting parameter-aware adjustments to ± 0.5 around the baseline. The initialization offset $\delta_k^\star$ preserves the tensor-wise mean coefficient at k_0 after clipping. P-DORM solves this offset once rather than treating it as a tunable hyperparameter. Appendix A gives the complete construction.

Optimizer-state reset. For each tensor l, we instantiate equation (8) with the tensor-specific statistics and k_l. When the resulting $\mathrm{Reset}_l^t = 1$, P-DORM preserves θ_l and clears only the corresponding momentum buffer:

$$u_l \leftarrow 0. \tag{11}$$

P-DORM also clears $\mathcal{W}_l$, excludes the triggering score from the new window, and retains the alignment statistics E_l and V_l. Algorithm 1 summarizes the SGD-based core. Appendix D gives the complete procedure and optimizer-specific extensions.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate P-DORM on CIFAR-10-C, CIFAR-100-C, and ImageNet-C (Hendrycks & Dietterich, 2019), and on the natural multi-domain benchmark DomainNet-126 (Peng et al., 2019; Döbler et al., 2023). The backbones include WideResNet-28-10 (Zagoruyko & Komodakis, 2016), an AugMix-pretrained ResNeXt-29 (Hendrycks et al., 2020; Xie et al., 2017), ResNet-50 (He et al., 2016), and ViT-B/16 (Dosovitskiy et al., 2021).

Test streams. We consider five settings with complementary distribution dynamics. **TTA** adapts to each target domain independently and reinitializes the model and optimizer before the next domain (Wang et al., 2021). **CTTA** presents abrupt domain shifts sequentially without reinitialization, testing whether adaptation remains stable as history accumulates (Wang et al., 2022). **GTTA** replaces abrupt severity changes with smooth transitions and evaluates tracking under gradual drift (Döbler et al., 2023). **PTTA** combines sequential domain shifts with temporally correlated labels and changing class proportions (Gong et al., 2022; Yuan et al., 2023). **Mixed** randomly interleaves samples from different domains, removing stable domain blocks and producing rapidly varying batch composition (Marsden et al., 2024). Appendix E gives the complete stream construction and domain order.

Baselines. Baselines include Source-only, TENT (Wang et al., 2021), DeYO (Lee et al., 2024), EATA-C (Tan et al., 2025), CoME (Zhang et al., 2025), BufferTTA (Kim et al., 2025), and RDumb (Press et al., 2023). For each adaptive method, we compare the unmodified optimizer with the corresponding P-DORM variant under both Adam and SGD (Kingma & Ba, 2015; Sutskever et al., 2013).

4.2 Main Results

Overall performance and compatibility. Tables 2–4 compare each unmodified optimizer (Pure) with the corresponding P-DORM variant. For every reported combination of dataset, backbone, and optimizer, we average Δ across methods. P-DORM improves all ten resulting averages, with reductions of 0.05–0.48 percentage points. Among the 60 reported method-level averages, 55 improve, one is unchanged at the reported precision, and the remaining four increase by at most 0.09 points. The consistent results across Adam and SGD, CNN and Transformer backbones, and diverse adaptation objectives demonstrate the broad compatibility of P-DORM.

Table 2: **Main results on CIFAR-10-C and CIFAR-100-C.** Five-seed mean error (%). Source-only directly evaluates the pretrained source model without adaptation. For each setting, $\Delta = \mathrm{Err}(\text{P-DORM}) - \mathrm{Err}(\text{Pure})$ on unrounded values, and Avg. Δ denotes the mean across settings. Green marks non-increasing error ($\Delta \leq 0$), and bold values indicate strict improvements. Gray marks error increases. The final row of each dataset and optimizer block averages Δ across methods.

	Method	Variant	Adam						SGD					
			TTA	CTTA	GTTA	PTTA	Mix	Avg.	TTA	CTTA	GTTA	PTTA	Mix	Avg.
(a) CIFAR-10-C	Source-only	–	43.52	43.52	24.66	43.52	43.52	39.75	43.52	43.52	24.66	43.52	43.52	39.75
	TENT (Wang et al., 2021)	Pure	19.14	18.53	12.23	66.06	32.12	29.62	19.61	18.39	12.28	67.74	32.53	30.11
		P-DORM	19.14	18.46	11.89	66.05	32.05	29.52	19.65	18.15	12.14	66.89	32.66	29.90
		Δ	0.00	**-0.07**	**-0.34**	**-0.01**	**-0.07**	**-0.10**	+0.04	**-0.24**	**-0.14**	**-0.85**	+0.13	**-0.21**
	DeYO (Lee et al., 2024)	Pure	19.20	17.89	12.23	65.83	28.85	28.80	20.02	17.88	12.31	66.34	28.06	28.92
		P-DORM	19.21	17.99	11.92	65.79	28.44	28.67	20.03	17.73	12.11	65.99	28.31	28.83
		Δ	+0.01	+0.10	**-0.31**	**-0.04**	**-0.41**	**-0.13**	+0.01	**-0.15**	**-0.20**	**-0.35**	+0.25	**-0.09**
	CoME (Zhang et al., 2025)	Pure	18.79	19.12	12.41	65.62	31.77	29.54	20.28	18.61	12.16	66.43	31.21	29.74
		P-DORM	18.80	18.84	12.17	65.54	31.46	29.36	20.28	18.59	12.22	65.90	31.44	29.69
		Δ	+0.01	**-0.28**	**-0.24**	**-0.08**	**-0.31**	**-0.18**	0.00	**-0.02**	+0.06	**-0.53**	+0.23	**-0.05**
	BufferTTA (Kim et al., 2025)	Pure	19.91	19.80	12.69	66.10	33.69	30.44	20.46	19.92	13.31	66.37	34.25	30.86
		P-DORM	19.91	19.66	12.43	66.09	33.64	30.35	20.46	19.41	12.44	66.08	33.53	30.38
		Δ	0.00	**-0.14**	**-0.26**	**-0.01**	**-0.05**	**-0.09**	0.00	**-0.51**	**-0.87**	**-0.29**	**-0.72**	**-0.48**
	EATA-C (Tan et al., 2025)	Pure	18.95	18.93	12.64	65.71	30.17	29.28	20.20	18.67	12.90	65.81	28.39	29.19
		P-DORM	18.94	18.75	12.40	65.63	29.79	29.10	20.21	18.59	12.73	65.54	28.80	29.17
		Δ	**-0.01**	**-0.18**	**-0.24**	**-0.08**	**-0.38**	**-0.18**	+0.01	**-0.08**	**-0.17**	**-0.27**	+0.41	**-0.02**
	RDumb (Press et al., 2023)	Pure	18.32	17.74	11.55	65.98	29.02	28.52	18.28	18.64	11.56	66.78	27.84	28.62
		P-DORM	18.33	17.74	11.52	65.98	29.00	28.51	18.32	17.94	11.42	66.46	28.08	28.45
		Δ	+0.01	0.00	**-0.03**	0.00	**-0.02**	**-0.01**	+0.04	**-0.70**	**-0.14**	**-0.32**	+0.24	**-0.17**
	Across methods	Avg. Δ	0.00	**-0.10**	**-0.24**	**-0.04**	**-0.21**	**-0.11**	+0.02	**-0.28**	**-0.24**	**-0.44**	+0.09	**-0.17**
(b) CIFAR-100-C	Source-only	–	46.45	46.45	33.58	46.45	46.45	43.88	46.45	46.45	33.58	46.45	46.45	43.88
	TENT (Wang et al., 2021)	Pure	31.82	32.74	27.45	42.52	44.79	35.86	35.16	33.20	27.95	41.95	45.77	36.81
		P-DORM	31.80	32.19	26.69	42.10	44.37	35.43	35.18	32.87	27.86	42.00	46.01	36.78
		Δ	**-0.02**	**-0.55**	**-0.76**	**-0.42**	**-0.42**	**-0.43**	+0.02	**-0.33**	**-0.09**	+0.05	+0.24	**-0.02**
	DeYO (Lee et al., 2024)	Pure	31.38	32.87	26.44	41.50	39.54	34.34	34.61	32.62	27.38	41.59	38.72	34.98
		P-DORM	31.38	32.36	26.18	41.09	38.69	33.94	34.64	32.14	27.24	41.42	39.32	34.95
		Δ	0.00	**-0.51**	**-0.26**	**-0.41**	**-0.85**	**-0.40**	+0.03	**-0.48**	**-0.14**	**-0.17**	+0.60	**-0.03**
	CoME (Zhang et al., 2025)	Pure	32.11	32.76	27.53	41.32	41.53	35.05	35.29	33.15	29.01	42.38	43.65	36.70
		P-DORM	32.11	32.41	27.09	41.19	41.48	34.86	35.30	33.41	28.64	42.20	43.59	36.63
		Δ	0.00	**-0.35**	**-0.44**	**-0.13**	**-0.05**	**-0.19**	+0.01	+0.26	**-0.37**	**-0.18**	**-0.06**	**-0.07**
	BufferTTA (Kim et al., 2025)	Pure	33.92	34.07	28.59	43.05	45.41	37.01	35.32	35.32	29.67	43.20	45.67	37.84
		P-DORM	33.92	33.86	28.00	43.01	45.38	36.83	35.32	35.31	29.89	43.20	45.67	37.88
		Δ	0.00	**-0.21**	**-0.59**	**-0.04**	**-0.03**	**-0.17**	0.00	**-0.01**	+0.22	0.00	0.00	+0.04
	EATA-C (Tan et al., 2025)	Pure	32.10	32.76	27.53	40.94	39.39	34.54	35.08	32.87	28.01	41.05	39.84	35.37
		P-DORM	32.11	32.34	26.87	40.59	38.84	34.15	35.10	32.64	27.81	40.98	39.42	35.19
		Δ	+0.01	**-0.42**	**-0.66**	**-0.35**	**-0.55**	**-0.39**	+0.02	**-0.23**	**-0.20**	**-0.07**	**-0.42**	**-0.18**
	RDumb (Press et al., 2023)	Pure	30.81	31.37	25.57	41.90	42.16	34.36	32.68	33.06	26.20	41.99	42.03	35.19
		P-DORM	30.80	31.31	25.62	41.77	41.95	34.29	32.80	32.48	26.17	41.65	42.31	35.08
		Δ	**-0.01**	**-0.06**	+0.05	**-0.13**	**-0.21**	**-0.07**	+0.12	**-0.58**	**-0.03**	**-0.34**	+0.28	**-0.11**
	Across methods	Avg. Δ	0.00	**-0.35**	**-0.44**	**-0.25**	**-0.35**	**-0.28**	+0.03	**-0.23**	**-0.10**	**-0.12**	+0.11	**-0.06**

Performance across test streams. P-DORM lowers the error averaged across methods in all ten CTTA and PTTA blocks. Under GTTA, it improves seven of the eight reported blocks; the exception is ViT-B/16 with SGD, whose method-averaged error increases by 0.10 points. Performance under TTA is nearly unchanged because the model, optimizer, and detector are reinitialized before each target domain, leaving little stale cross-domain history to remove. This result confirms that P-DORM remains stable when an optimizer-state reset is unnecessary. Under the Mixed setting, P-DORM improves seven blocks and matches one; the only increases occur with SGD on the smaller-scale CIFAR-10-C and CIFAR-100-C benchmarks. Their shorter effective adaptation trajectories provide less history for separating transient mixture noise from stale domain-specific momentum, so resets may occasionally discard useful gradient averaging.

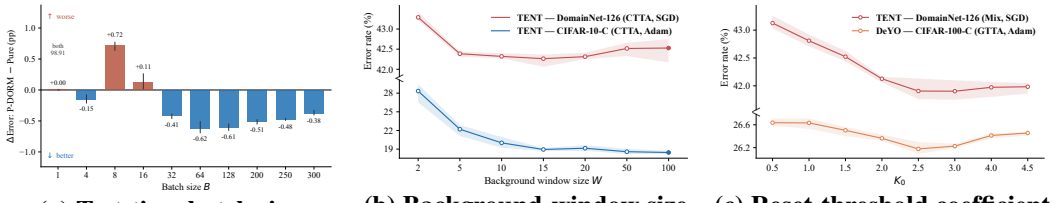
4.3 Sensitivity Analysis

Batch size B. Figure 2(a) evaluates DeYO with Adam on CIFAR-100-C. P-DORM provides stable gains for $B \geq 32$ but is less consistent with very small batches, whose noisy gradients weaken the gradient–state alignment signal. Extremely small-batch adaptation therefore remains a limitation.

Background-window size W. Figure 2(b) varies W for TENT on CIFAR-10-C and DomainNet-126 under the CTTA setting. We set $W_{\min} = W$ for $W < 20$ and $W_{\min} = 15$ otherwise. Very short windows provide noisy background estimates and degrade performance, whereas moderate and large windows remain stable. The default $W = 100$ performs strongly on both datasets.

Table 3: **Main results on ImageNet-C with ResNet-50 and ViT-B/16.** Five-seed mean error (%). Notation and highlighting follow Table 2.

	Method	Variant	Adam						SGD					
			TTA	CTTA	GTTA	PTTA	Mix	Avg.	TTA	CTTA	GTTA	PTTA	Mix	Avg.
(a) ResNet-50	Source-only	–	82.02	82.02	58.41	82.02	82.02	77.30	82.02	82.02	58.41	82.02	82.02	77.30
	TENT (Wang et al., 2021)	Pure	68.39	64.54	44.97	65.15	82.31	65.07	65.52	63.48	45.16	64.36	83.66	64.44
		P-DORM	68.39	63.83	44.45	64.58	82.27	64.71	65.75	63.29	44.92	64.12	83.50	64.31
		Δ	0.00	**-0.71**	**-0.52**	**-0.57**	**-0.04**	**-0.37**	+0.23	**-0.19**	**-0.24**	**-0.24**	**-0.16**	**-0.12**
	DeYO (Lee et al., 2024)	Pure	68.13	65.86	43.52	66.42	80.32	64.85	67.48	64.05	43.82	65.29	79.25	63.98
		P-DORM	68.12	64.90	43.64	65.57	79.72	64.39	67.52	63.91	43.90	64.74	78.56	63.73
		Δ	**-0.01**	**-0.96**	+0.12	**-0.85**	**-0.60**	**-0.46**	+0.04	**-0.14**	+0.08	**-0.55**	**-0.69**	**-0.25**
	CoME (Zhang et al., 2025)	Pure	68.44	64.93	45.10	65.42	81.80	65.14	65.13	63.97	46.26	64.66	80.69	64.14
		P-DORM	68.43	64.23	44.59	64.78	81.55	64.71	65.29	63.37	44.95	64.05	80.58	63.65
		Δ	**-0.01**	**-0.70**	**-0.51**	**-0.64**	**-0.25**	**-0.42**	+0.16	**-0.60**	**-1.31**	**-0.61**	**-0.11**	**-0.49**
	BufferTTA (Kim et al., 2025)	Pure	68.12	67.38	43.44	68.08	83.64	66.13	68.60	68.49	44.62	69.11	83.00	66.77
		P-DORM	68.12	67.43	43.08	67.92	83.82	66.07	68.59	68.31	44.71	68.79	82.70	66.62
		Δ	0.00	+0.05	**-0.36**	**-0.16**	+0.18	**-0.06**	**-0.01**	**-0.18**	+0.09	**-0.32**	**-0.30**	**-0.14**
	EATA-C (Tan et al., 2025)	Pure	68.20	66.68	44.66	67.25	79.93	65.34	67.21	65.57	46.90	66.21	76.88	64.56
		P-DORM	68.19	66.23	45.50	66.87	79.24	65.21	67.25	65.35	45.98	65.99	76.76	64.27
		Δ	**-0.01**	**-0.45**	+0.84	**-0.38**	**-0.69**	**-0.14**	+0.04	**-0.22**	**-0.92**	**-0.22**	**-0.12**	**-0.29**
	RDumb (Press et al., 2023)	Pure	62.13	60.03	40.52	60.99	81.72	61.08	61.38	61.65	41.65	62.65	79.03	61.27
		P-DORM	62.13	59.68	40.53	60.67	82.28	61.06	61.32	61.20	41.39	62.10	79.09	61.02
		Δ	0.00	**-0.35**	+0.01	**-0.32**	+0.56	**-0.02**	**-0.06**	**-0.45**	**-0.26**	**-0.55**	+0.06	**-0.25**
	Across methods	Avg. Δ	0.00	**-0.52**	**-0.07**	**-0.49**	**-0.14**	**-0.24**	+0.07	**-0.30**	**-0.43**	**-0.41**	**-0.22**	**-0.26**
(b) ViT-B/16	Source-only	–	53.98	53.98	35.10	53.98	53.98	50.20	53.98	53.98	35.10	53.98	53.98	50.20
	TENT (Wang et al., 2021)	Pure	42.98	44.29	28.58	44.25	43.04	40.63	44.99	45.63	27.98	45.68	43.33	41.52
		P-DORM	43.10	43.31	28.28	43.26	42.76	40.14	45.02	44.78	28.16	44.55	43.58	41.22
		Δ	+0.12	**-0.98**	**-0.30**	**-0.99**	**-0.28**	**-0.49**	+0.03	**-0.85**	+0.18	**-1.13**	+0.25	**-0.30**
	DeYO (Lee et al., 2024)	Pure	41.63	44.78	27.91	43.41	43.10	40.17	47.49	46.51	27.73	44.34	44.26	42.07
		P-DORM	41.64	43.10	27.55	42.33	42.68	39.46	47.61	43.97	27.96	44.29	43.11	41.39
		Δ	+0.01	**-1.68**	**-0.36**	**-1.08**	**-0.42**	**-0.71**	+0.12	**-2.54**	+0.23	**-0.05**	**-1.15**	**-0.68**
	CoME (Zhang et al., 2025)	Pure	44.95	44.67	28.70	44.76	44.66	41.55	44.60	46.13	28.03	46.03	45.01	41.96
		P-DORM	44.79	42.48	27.98	42.55	43.67	40.30	44.15	45.93	28.34	45.77	44.74	41.79
		Δ	**-0.16**	**-2.19**	**-0.72**	**-2.21**	**-0.99**	**-1.25**	**-0.45**	**-0.20**	+0.31	**-0.26**	**-0.27**	**-0.17**
	BufferTTA (Kim et al., 2025)	Pure	50.32	49.75	28.99	49.80	48.58	45.49	52.55	50.40	28.59	46.57	46.98	45.02
		P-DORM	50.19	49.58	28.59	49.64	48.68	45.34	51.78	45.68	28.68	47.06	45.84	43.81
		Δ	**-0.13**	**-0.17**	**-0.40**	**-0.16**	+0.10	**-0.15**	**-0.77**	**-4.72**	+0.09	+0.49	**-1.14**	**-1.21**
	EATA-C (Tan et al., 2025)	Pure	42.22	42.16	27.98	42.27	43.30	39.59	46.85	44.41	28.13	44.60	44.18	41.63
		P-DORM	42.22	41.51	27.74	41.51	43.02	39.20	46.88	43.94	27.85	44.38	43.47	41.30
		Δ	0.00	**-0.65**	**-0.24**	**-0.76**	**-0.28**	**-0.38**	+0.03	**-0.47**	**-0.28**	**-0.22**	**-0.71**	**-0.33**
	RDumb (Press et al., 2023)	Pure	38.18	38.90	25.73	39.48	42.36	36.93	44.37	42.03	27.11	42.92	44.77	40.24
		P-DORM	38.17	39.23	25.85	39.33	42.49	37.02	44.50	41.55	27.17	42.46	44.59	40.05
		Δ	**-0.01**	+0.33	+0.12	**-0.15**	+0.13	+0.09	+0.13	**-0.48**	+0.06	**-0.46**	**-0.18**	**-0.19**
	Across methods	Avg. Δ	**-0.03**	**-0.89**	**-0.32**	**-0.89**	**-0.29**	**-0.48**	**-0.15**	**-1.54**	+0.10	**-0.27**	**-0.53**	**-0.48**



(a) Test-time batch size B **(b) Background-window size** W **(c) Reset-threshold coefficient** k_0

Figure 2: **Sensitivity to batch size, window size, and reset threshold.** (a) DeYO on CIFAR-100-C (CTTA). (b) TENT under two representative CTTA configurations. (c) DeYO on CIFAR-100-C (GTTA) and TENT on DomainNet-126 (Mixed). Panel (a) reports the paired error difference defined in Table 2, while panels (b)–(c) report P-DORM error. Whiskers and bands denote seed-wise ranges.

Reset-threshold coefficient k_0. Figure 2(c) varies k_0 with clipping bounds $[k_0 - 0.5, k_0 + 0.5]$. Both DeYO on CIFAR-100-C under the GTTA setting and TENT on DomainNet-126 under the Mixed setting perform best at intermediate values of k_0. A small k_0 lowers the threshold and causes overly frequent resets, which can discard useful optimizer history. A large k_0 suppresses resets and may retain stale history for too long. Intermediate values therefore provide the most effective reset frequency in these settings.

Parameter-aware allocation shape. To validate the Parameter-Aware Threshold Allocation in Section 3.3, we compare five allocation shapes in Table 5(a). Fixed assigns equal sensitivity to all module sizes. Small, the P-DORM default, favors smaller modules. Large favors larger modules,

Table 4: **Main results on DomainNet-126 with ResNet-50.** Five-seed mean error (%). GTTA is omitted because DN-126 has no ordered severity axis for constructing gradual transitions. Notation and highlighting follow Table 2.

Method	Variant	Adam					SGD				
		TTA	CTTA	PTTA	Mix	Avg.	TTA	CTTA	PTTA	Mix	Avg.
Source-only	–	45.28	45.28	45.28	44.67	45.13	45.28	45.28	45.28	44.67	45.13
TENT (Wang et al., 2021)	Pure	42.51	42.68	54.24	42.18	45.40	42.53	43.23	54.72	42.00	45.62
	P-DORM	42.47	42.47	54.17	42.05	45.29	42.57	42.53	54.35	41.91	45.34
	Δ	**-0.04**	**-0.21**	**-0.07**	**-0.13**	**-0.11**	+0.04	**-0.70**	**-0.37**	**-0.09**	**-0.28**
DeYO (Lee et al., 2024)	Pure	44.94	44.26	54.90	44.15	47.06	45.41	44.64	55.79	44.97	47.70
	P-DORM	44.85	43.97	54.89	44.13	46.96	45.35	44.67	55.56	44.73	47.58
	Δ	**-0.09**	**-0.29**	**-0.01**	**-0.02**	**-0.10**	**-0.06**	+0.03	**-0.23**	**-0.24**	**-0.12**
CoME (Zhang et al., 2025)	Pure	40.64	41.36	53.77	40.94	44.18	43.91	42.20	53.93	41.42	45.37
	P-DORM	40.69	41.42	53.72	41.16	44.25	44.03	42.03	53.92	41.23	45.30
	Δ	+0.06	+0.05	**-0.06**	+0.22	+0.07	+0.12	**-0.17**	0.00	**-0.19**	**-0.06**
BufferTTA (Kim et al., 2025)	Pure	45.72	45.63	55.14	45.31	47.95	45.76	45.77	55.22	45.54	48.07
	P-DORM	45.72	45.62	55.17	45.34	47.96	45.76	45.76	55.22	45.54	48.07
	Δ	0.00	**-0.02**	+0.03	+0.03	+0.01	0.00	**-0.01**	0.00	0.00	0.00
EATA-C (Tan et al., 2025)	Pure	42.92	42.01	53.36	42.34	45.16	43.08	42.88	54.54	41.87	45.59
	P-DORM	42.88	41.79	53.31	42.23	45.05	43.16	42.67	54.16	41.88	45.47
	Δ	**-0.04**	**-0.22**	**-0.05**	**-0.11**	**-0.11**	+0.08	**-0.21**	**-0.38**	+0.01	**-0.12**
RDumb (Press et al., 2023)	Pure	43.01	41.95	53.75	41.73	45.11	42.78	44.53	54.56	42.16	46.01
	P-DORM	42.97	41.74	53.84	41.74	45.07	42.69	43.70	54.42	42.07	45.72
	Δ	**-0.04**	**-0.21**	+0.09	+0.01	**-0.04**	**-0.09**	**-0.83**	**-0.15**	**-0.09**	**-0.29**
Across methods	Avg. Δ	**-0.02**	**-0.15**	**-0.01**	0.00	**-0.05**	+0.02	**-0.31**	**-0.19**	**-0.10**	**-0.15**

Table 5: **Additional validation.** (a) Error rates (%) for five threshold-allocation shapes. Small is the five-seed default. (b) Five-seed supervised non-stationary training error (%) on DomainNet-126.

(a) Parameter-aware allocation shapes

Setting	Fixed	Large	Mid	Ends	Small
TENT/C10C/CTTA	18.85	19.62	19.36	19.48	**18.46**
DeYO/C100C/CTTA	32.42	32.46	32.37	32.47	**32.36**
CoME/C100C/Mixed	41.87	42.39	41.77	41.99	**41.48**
DeYO/C100C/GTTA	26.40	26.57	26.37	26.43	**26.18**

(b) Supervised non-stationary training

	Stage		Final	
Method	Error	Δ	Error	Δ
Pure	43.70	–	47.16	–
P-DORM	42.32	**-1.38**	43.90	**-3.25**
Boundary-Oracle	39.00	**-4.70**	40.42	**-6.74**

Mid favors intermediate sizes, and Ends favors both size extremes. Appendix C details all five variants. Small achieves the lowest error in all four reported configurations, supporting parameter-aware allocation.

Supervised non-stationary training. To test whether P-DORM extends beyond unlabeled TTA, we construct a labeled non-stationary DomainNet-126 stream from sequential target-domain shards, as detailed in Appendix F.3. Table 5(b) compares Pure, P-DORM, and Boundary-Oracle. P-DORM lowers both stage and final errors, showing that optimizer-state reset also benefits supervised non-stationary training.

5 Conclusion

We identify stale optimizer history as an overlooked source of adaptation lag in non-stationary TTA and introduce P-DORM, a plug-and-play wrapper that resets incompatible optimizer states without reverting adapted model parameters. P-DORM measures tensor-wise misalignment between the current gradient and the optimizer's historical direction, calibrates this signal using tensor-specific online statistics, and applies parameter-aware thresholds to trigger tensor-specific resets. We prove that raw directional alignment is not statistically comparable across tensors because its center and fluctuation scale depend on tensor dimension and directional geometry, which motivates the proposed standardization. Extensive experiments across TTA methods, optimizers, architectures, and test streams demonstrate improved overall adaptation performance, with clearer gains when optimizer history persists across distribution shifts. These results establish optimizer-state management as an important component of non-stationary TTA and P-DORM as a practical approach to robust adaptation.

AI USE STATEMENT

Generative AI tools were used to aid and polish the writing of this paper, including language editing and presentation refinement. The mathematical arguments and technical content were developed by the authors. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final manuscript.

REFERENCES

Kavosh Asadi, Rasool Fakoor, and Shoham Sabach. Resetting the optimizer in deep RL: An empirical study. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 72284–72324, 2023.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

Niladri S. Chatterji, Behnam Neyshabur, and Hanie Sedghi. The intriguing role of module criticality in the generalization of deep networks. In *International Conference on Learning Representations (ICLR)*, 2020.

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

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 49205–49233, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Qi Deng, Shuaicheng Niu, Ronghao Zhang, Yaofo Chen, Runhao Zeng, Jian Chen, and Xiping Hu. Learning to generate gradients for test-time adaptation via test-time training layers. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 16235–16243, 2025.

Mario Döbler, Robert A. Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7704–7714, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. NOTE: Robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 27253–27266, 2022.

Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, 2014. doi: 10.1109/TKDE.2013.184.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations (ICLR)*, 2020.

Shaoyang Huang, Yashi Zhu, Yichen Yu, Lei Zhang, Zhang Yi, and Tao He. BP-TTA: Balanced and prototype-guided test-time adaptation in dynamic scenarios. *arXiv preprint arXiv:2606.31420*, 2026.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, volume 37, pp. 448–456, 2015.

Hyeongyu Kim, Geonhui Han, and Dosik Hwang. Buffer layers for test-time adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 141127–141149, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, pp. 5637–5664, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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

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

Kaizhao Liang, Lizhang Chen, Bo Liu, and Qiang Liu. Cautious optimizers: Improving training with one line of code. In *International Conference on Learning Representations (ICLR)*, 2026.

Taejun Lim, Joong-Won Hwang, and Kibok Lee. When and where to reset matters for long-term test-time adaptation. In *International Conference on Learning Representations (ICLR)*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Ke Ma, Jiaqi Tang, Bin Guo, Fan Dang, Sicong Liu, Zhui Zhu, Lei Wu, Cheng Fang, Ying-Cong Chen, Zhiwen Yu, and Yunhao Liu. SURGEON: Memory-adaptive fully test-time adaptation via dynamic activation sparsity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30514–30523, 2025.

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 (WACV)*, pp. 2555–2565, 2024.

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 *International Conference on Learning Representations (ICLR)*, 2023.

Brendan O'Donoghue and Emmanuel Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, 2015.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1406–1415, 2019.

Ori Press, Steffen Schneider, Matthias Kümmerer, and Matthias Bethge. RDumb: A simple approach that questions our progress in continual test-time adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 39915–39935, 2023.

PyTorch Contributors. RMSprop: PyTorch documentation. `https://docs.pytorch.org/docs/stable/generated/torch.optim.rmsprop.RMSprop.html`, 2026. Accessed September 4, 2026. Includes the momentum variant.

Sharan Sahu, Cameron J. Hogan, and Martin T. Wells. On the provable suboptimality of momentum SGD in nonstationary stochastic optimization. In *International Conference on Machine Learning (ICML)*, 2026.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning (ICML)*, pp. 1139–1147, 2013.

Mingkui Tan, Guohao Chen, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Peilin Zhao, and Shuaicheng Niu. Uncertainty-calibrated test-time model adaptation without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE TPAMI)*, 47(8):6274–6289, 2025. doi: 10.1109/TPAMI.2025.3560696.

Kristi Topollai and Anna Ewa Choromanska. Adaptive memory momentum via a model-based framework for deep learning optimization. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 300 of *Proceedings of Machine Learning Research*, pp. 4060–4068, 2026.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. TENT: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

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

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1492–1500, 2017.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15922–15932, 2023.

Yige Yuan, Bingbing Xu, Liang Hou, Fei Sun, Huawei Shen, and Xueqi Cheng. TEA: Test-time energy adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23901–23911, 2024.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*, pp. 87.1–87.12, 2016. doi: 10.5244/C.30.87.

Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal? *Journal of Machine Learning Research (JMLR)*, 23(67):1–28, 2022.

Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. COME: Test-time adaption by conservatively minimizing entropy. In *International Conference on Learning Representations (ICLR)*, 2025.

Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 18795–18806, 2020.

A PARAMETER-AWARE THRESHOLD CALIBRATION

The size rank in equation (9) is defined over modules, whereas the reset rule is applied to parameter tensors. Because modules may contain different numbers of trainable tensors, a zero-mean allocation across modules need not have zero mean across tensors. Clipping in equation (10) can introduce an additional shift. We therefore calibrate one architecture-dependent offset so that heterogeneous allocation changes relative sensitivities without changing the tensor-wise mean of the coefficients.

The implementation first groups tensors by removing the `weight`/`bias` suffix from parameter names and defines module size by the largest tensor in each module:

$$n_m = \max_{l:\, m(l)=m} \operatorname{numel}(\theta_l). \tag{12}$$

Assume $M \geq 2$. Let $\operatorname{lo}(n_m)$ and $\operatorname{hi}(n_m)$ be the zero-based left and right insertion positions of n_m in the sorted module-size sequence. Sorting by n_m is equivalent to sorting by $\log n_m$ in equation (9) because the logarithm is strictly increasing. Tied sizes use their average rank:

$$\bar{r}_m = \frac{\operatorname{lo}(n_m) + \operatorname{hi}(n_m) + 1}{2}, \qquad w_m = \frac{\bar{r}_m - 1}{M - 1}. \tag{13}$$

The shape function and uncalibrated module coefficient are

$$s(w_m) = 2w_m - 1, \qquad \widetilde{k}_m = k_0 + \lambda s(w_m). \tag{14}$$

The default choice, termed *Small*, assigns lower coefficients to smaller modules and therefore makes them easier to reset. Appendix C defines the alternative allocation shapes used in the ablation.

Here, n_m is the number of elements in the largest trainable tensor of module m, not the module's total parameter count. If all monitored modules have the same size, average ranking gives $w_m = 1/2$ for every module. The default shape then satisfies $s(w_m) = 0$, and mean-preserving calibration yields $k_l = k_0$ for every tensor. The parameter-aware allocation therefore reduces exactly to a uniform threshold in this case. This occurs for the equally sized LayerNorm affine modules monitored in our ViT-B/16 configurations, so their gains arise from detection and state reset rather than size-dependent allocation.

A.1 MEAN-PRESERVING CALIBRATION

Let c_m denote the number of parameter tensors in module m, so that $L = \sum_m c_m$. Define the monotone non-decreasing function

$$\begin{aligned} F(\delta) &= \frac{1}{L} \sum_{m=1}^{M} c_m \, \operatorname{clip}\Big(\widetilde{k}_m + \delta, k_{\min}, k_{\max}\Big) \\ &= \frac{1}{L} \sum_{l=1}^{L} \operatorname{clip}\big(k_0 + \lambda s(w_{m(l)}) + \delta, k_{\min}, k_{\max}\big)\,. \end{aligned} \tag{15}$$

The factor c_m makes the module-wise expression exactly equal to the tensor-wise mean. The calibration $\delta_k^\star$ is any solution of $F(\delta_k^\star) = k_0$. Since clipping is continuous and monotone in δ, so is F. With $k_{\min} \leq k_0 \leq k_{\max}$, sufficiently negative and positive offsets yield $F(\delta) = k_{\min}$ and $F(\delta) = k_{\max}$, respectively. Hence, a solution exists by continuity. Starting from a bracket containing a solution, the implementation uses bisection:

$$\begin{aligned} q_j &= \frac{\ell_j + u_j}{2}, \\ (\ell_{j+1}, u_{j+1}) &= \begin{cases} (q_j, u_j), & F(q_j) < k_0, \\ (\ell_j, q_j), & F(q_j) \geq k_0, \end{cases} \\ \delta_k^\star &= \frac{\ell_{40} + u_{40}}{2}. \end{aligned} \tag{16}$$

The subscript 40 denotes the fixed 40 bisection iterations used to obtain numerical convergence. We set $k_{\min} = k_0 - 0.5$ and $k_{\max} = k_0 + 0.5$. The default $k_0 = 2.5$ therefore gives $[k_{\min}, k_{\max}] =$

[2, 3]. Together with $\lambda \leq 0.5$, these bounds ensure that $[-1, 1]$ brackets a solution. Flat regions introduced by clipping may make the solution non-unique, but every solution enforces the same mean constraint. Each tensor finally receives $k_l = \mathrm{clip}(\widetilde{k}_{m(l)} + \delta_k^\star, k_{\min}, k_{\max})$. Both $\delta_k^\star$ and $\{k_l\}_{l=1}^L$ are computed once at initialization and cached. Thus, $\delta_k^\star$ is a calibration result rather than a tunable hyperparameter. This correction preserves only the mean coefficient. The realized thresholds and reset frequencies still depend on the background statistics and anomaly sequence of each tensor.

B Dimension-Dependent Fluctuations of Directional Alignment

This appendix explains why raw cosine alignment is not statistically comparable across parameter tensors and why P-DORM standardizes it online. We begin with an exact isotropic null model in Section B.1, extend the analysis to correlated and anisotropic directions in Sections B.2–B.3, and analyze the online P-DORM score in Section B.4. Section B.5 then defines the raw-alignment control used to isolate the effect of this standardization. Each result is stated under explicit assumptions. In particular, dimension alone does not determine an observed alignment.

B.1 Exact Null Fluctuations under Spherical Symmetry

Consider parameter tensor l, and let $d = d_l \geq 2$ denote the number of elements in the tensor. Let $P, Q \in \mathbb{R}^d$ be independent, almost surely nonzero, spherically symmetric random vectors. Define their null cosine alignment by

$$C_d = \frac{P^\top Q}{\|P\|_2 \|Q\|_2}. \tag{17}$$

Proposition B.1. Under these assumptions,

$$\mathbb{E}[C_d] = 0, \qquad \mathrm{Var}(C_d) = \frac{1}{d}. \tag{18}$$

Proof. Set $U = P/\|P\|_2$ and $V = Q/\|Q\|_2$. Independence and spherical symmetry imply that U and V are independent and uniform on the unit sphere $\mathbb{S}^{d-1}$. Let $e_1 = (1, 0, \ldots, 0)^\top$ be the first canonical basis vector. Conditional on any fixed V, choose an orthogonal matrix R satisfying $RV = e_1$. Rotational invariance gives $RU \overset{d}{=} U$, and therefore

$$C_d = U^\top V = (RU)^\top (RV) = (RU)_1 \overset{d}{=} U_1. \tag{19}$$

Antipodal symmetry gives $\mathbb{E}[U_1] = 0$. Moreover, $\sum_{i=1}^d U_i^2 = 1$ almost surely, and exchangeability of the coordinates yields

$$1 = \sum_{i=1}^d \mathbb{E}[U_i^2] = d\,\mathbb{E}[U_1^2]. \tag{20}$$

Thus $\mathbb{E}[U_1^2] = 1/d$. Equation (19) then gives both claims. □

Proposition B.1 is exact for every $d \geq 2$: an unaligned isotropic cosine has standard deviation $d^{-1/2}$. Hence, the same absolute alignment change has different null significance at different tensor dimensions.

B.2 Correlated Gaussian Directions

The null model above has zero expected alignment. To model a nonzero within-domain baseline, let $X = (X_1, \ldots, X_d)^\top$ and $Y = (Y_1, \ldots, Y_d)^\top$, where the coordinate pairs $\{(X_i, Y_i)\}_{i=1}^d$ are i.i.d. zero-mean jointly Gaussian and satisfy

$$\mathbb{E}[X_i^2] = \mathbb{E}[Y_i^2] = 1, \qquad \mathbb{E}[X_i Y_i] = \rho, \qquad \rho \in (-1, 1) \text{ fixed.} \tag{21}$$

Here ρ is the population correlation between corresponding coordinates. In P-DORM, ρ idealizes the normal within-domain alignment between a tensor's current gradient and the corresponding pre-update optimizer direction. Unit marginal variances are without loss of generality because cosine

alignment is unchanged by separate positive rescalings. Define

$$C_d = \frac{X^\top Y}{\|X\|_2 \|Y\|_2}. \tag{22}$$

For completeness, introduce

$$\eta_i = \frac{Y_i - \rho X_i}{\sqrt{1-\rho^2}}. \tag{23}$$

Because (X_i, Y_i) is jointly Gaussian, (X_i, η_i) is also jointly Gaussian. Equation (21) gives $\mathbb{E}[\eta_i] = 0$, $\mathrm{Var}(\eta_i) = 1$, and $\mathrm{Cov}(X_i, \eta_i) = 0$. Joint Gaussianity therefore implies $X_i \perp \eta_i$. Thus the model has the equivalent representation

$$Y_i = \rho X_i + \sqrt{1-\rho^2}\,\eta_i, \qquad X_i, \eta_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \qquad X_i \perp \eta_i. \tag{24}$$

Proposition B.2. As $d \to \infty$,

$$C_d \xrightarrow{\text{a.s.}} \rho, \qquad \sqrt{d}\,(C_d - \rho) \xrightarrow{\mathcal{D}} \mathcal{N}\big(0, (1-\rho^2)^2\big)\,. \tag{25}$$

Moreover,

$$\mathrm{Var}(C_d) = \frac{(1-\rho^2)^2}{d} + o(d^{-1}). \tag{26}$$

Proof. Write the numerator and squared norms in equation (22) as empirical moments:

$$A_d = \frac{1}{d}\sum_{i=1}^{d} X_i Y_i, \quad B_d = \frac{1}{d}\sum_{i=1}^{d} X_i^2, \quad D_d = \frac{1}{d}\sum_{i=1}^{d} Y_i^2. \tag{27}$$

Indeed, $X^\top Y = dA_d$, $\|X\|_2^2 = dB_d$, and $\|Y\|_2^2 = dD_d$, so

$$C_d = \frac{dA_d}{\sqrt{dB_d}\sqrt{dD_d}} = h(A_d, B_d, D_d), \qquad h(a,b,c) = \frac{a}{\sqrt{bc}}. \tag{28}$$

The expectations in equation (21) give $\mathbb{E}(A_d, B_d, D_d) = (\rho, 1, 1)$. The strong law of large numbers and continuity of h at $(\rho, 1, 1)$ therefore imply $C_d \to h(\rho, 1, 1) = \rho$ almost surely.

For the fluctuation limit, let $W_i = (X_i Y_i, X_i^2, Y_i^2)^\top$. Isserlis' theorem gives

$$\mathbb{E}[X_i^2 Y_i^2] = 1 + 2\rho^2, \quad \mathbb{E}[X_i^3 Y_i] = \mathbb{E}[X_i Y_i^3] = 3\rho, \quad \mathbb{E}[X_i^4] = \mathbb{E}[Y_i^4] = 3. \tag{29}$$

Subtracting the corresponding products of first moments yields

$$\mathrm{Cov}(W_i) = \Sigma_\rho = \begin{pmatrix} 1+\rho^2 & 2\rho & 2\rho \\ 2\rho & 2 & 2\rho^2 \\ 2\rho & 2\rho^2 & 2 \end{pmatrix}. \tag{30}$$

The multivariate central limit theorem now gives

$$\sqrt{d}\left[\begin{pmatrix} A_d \\ B_d \\ D_d \end{pmatrix} - \begin{pmatrix} \rho \\ 1 \\ 1 \end{pmatrix}\right] \xrightarrow{\mathcal{D}} \mathcal{N}(0, \Sigma_\rho). \tag{31}$$

Since

$$\nabla h(\rho, 1, 1) = \left(1, -\frac{\rho}{2}, -\frac{\rho}{2}\right)^\top, \tag{32}$$

the delta method applied to equation (28) and equation (31) gives the second convergence in equation (25), with variance

$$\begin{aligned} \nabla h^\top \Sigma_\rho \nabla h &= (1+\rho^2) - 2\rho^2 - 2\rho^2 + \frac{\rho^2}{2} + \rho^4 + \frac{\rho^2}{2} \\ &= (1-\rho^2)^2, \end{aligned} \tag{33}$$

where $\nabla h = \nabla h(\rho, 1, 1)$.

Weak convergence alone does not ensure convergence of second moments, so we justify the variance expansion separately. Let $\xi_d = (A_d - \rho, B_d - 1, D_d - 1)^\top$ and $G_d = \{B_d, D_d \in [1/2, 3/2]\}$. Gaussian concentration gives $\Pr(G_d^c) \le \exp[-\Omega(d)]$. Also, Cauchy–Schwarz gives $A_d^2 \le B_d D_d$. Hence, the line segment between (A_d, B_d, D_d) and $(\rho, 1, 1)$ remains in a fixed compact set on G_d. The Hessian of h is bounded there, and Taylor's theorem yields

$$C_d - \rho = \nabla h(\rho, 1, 1)^\top \xi_d + R_d, \qquad |R_d| \le K \|\xi_d\|_2^2. \tag{34}$$

Because the W_i are i.i.d. with finite fourth moments, $\mathbb{E}\|\xi_d\|_2^4 = O(d^{-2})$. Together with $\mathbb{E}[\xi_d] = 0$ and $\mathrm{Cov}(\xi_d) = \Sigma_\rho / d$, this gives

$$\mathbb{E}\big[(\nabla h^\top \xi_d)^2\big] = \frac{1}{d} \nabla h^\top \Sigma_\rho \nabla h, \tag{35}$$

while the squared remainder contributes $O(d^{-2})$ and the cross term between the remainder and the linear part is $O(d^{-3/2})$ by Cauchy–Schwarz. Contributions from G_d^c are exponentially small because $|C_d - \rho| \le 2$. Consequently, $\mathbb{E}[C_d] - \rho = O(d^{-1})$ and

$$\begin{aligned} \mathrm{Var}(C_d) &= \mathbb{E}[(C_d - \rho)^2] - (\mathbb{E}[C_d] - \rho)^2 \\ &= \frac{1}{d} \nabla h(\rho, 1, 1)^\top \Sigma_\rho \nabla h(\rho, 1, 1) + o(d^{-1}) \\ &= \frac{(1 - \rho^2)^2}{d} + o(d^{-1}), \end{aligned} \tag{36}$$

which proves equation (26). □

Equation (26) shows why a raw alignment drop is not comparable across tensors. Even at the same normal correlation ρ, the first-order standard deviation of the alignment is $(1 - \rho^2)/\sqrt{d}$. Therefore, the same absolute drop has different null significance at different dimensions. To remove this center and scale dependence, define

$$Z_d^\star = \frac{\rho - C_d}{(1 - \rho^2)/\sqrt{d}} = -\frac{\sqrt{d}\,(C_d - \rho)}{1 - \rho^2} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1). \tag{37}$$

The convergence follows from Proposition B.2 and the continuous mapping theorem because $1 - \rho^2 > 0$. Thus, for any fixed $x \in \mathbb{R}$,

$$\Pr\left(\rho - C_d > \frac{(1 - \rho^2)x}{\sqrt{d}}\right) = \Pr(Z_d^\star > x) \longrightarrow 1 - \Phi(x), \tag{38}$$

where Φ is the standard-normal distribution function. A common threshold on $Z_d^\star$ therefore has a common asymptotic false-alarm interpretation, whereas a common raw threshold does not.

This is the statistical role of the first calibration stage in P-DORM: Section 3.2 centers and scales each tensor's alignment innovation. The window rule in Section 3.3 subsequently suppresses residual temporal noise and controls intervention sensitivity.

B.3 Anisotropic Fluctuations and Effective Dimension

We now relax the spherical-symmetry assumption of Section B.1. Let $P, Q \in \mathbb{R}^d$ be independent, centrally symmetric, almost surely nonzero random vectors with covariance matrices Σ_P and Σ_Q, and let C_d be defined by equation (17). Define the directional second moments

$$A_P = \mathbb{E}\left[\frac{PP^\top}{\|P\|_2^2}\right], \qquad A_Q = \mathbb{E}\left[\frac{QQ^\top}{\|Q\|_2^2}\right]. \tag{39}$$

Central symmetry gives $\mathbb{E}[P/\|P\|_2] = \mathbb{E}[Q/\|Q\|_2] = 0$. Independence therefore yields $\mathbb{E}[C_d] = 0$ and the exact identity

$$\mathrm{Var}(C_d) = \mathbb{E}[C_d^2] = \mathrm{tr}(A_P A_Q). \tag{40}$$

This identity shows directly that fluctuation depends on directional geometry. As a heuristic simplification, when both squared norms concentrate tightly around their expectations, replacing the

random denominators in equation (39) by $\operatorname{tr}(\Sigma_P)$ and $\operatorname{tr}(\Sigma_Q)$ gives $A_P \approx \Sigma_P/\operatorname{tr}(\Sigma_P)$ and $A_Q \approx \Sigma_Q/\operatorname{tr}(\Sigma_Q)$. Consequently,

$$\operatorname{Var}(C_d) \approx \frac{\operatorname{tr}(\Sigma_P\Sigma_Q)}{\operatorname{tr}(\Sigma_P)\operatorname{tr}(\Sigma_Q)}. \tag{41}$$

For the common-covariance case $\Sigma_P = \Sigma_Q = \Sigma$, define the participation-ratio effective dimension

$$d_{\text{eff}} = \frac{\operatorname{tr}(\Sigma)^2}{\operatorname{tr}(\Sigma^2)} = \frac{(\sum_{i=1}^d \lambda_i)^2}{\sum_{i=1}^d \lambda_i^2}, \tag{42}$$

where $\{\lambda_i\}$ are the eigenvalues of Σ. Equation (41) then becomes $\operatorname{Var}(C_d) \approx 1/d_{\text{eff}}$. Under isotropy, all eigenvalues are equal and $d_{\text{eff}} = d$. When variance concentrates in fewer directions, d_{eff} decreases. Thus even tensors with the same nominal dimension can have different alignment variances. Nominal parameter count is therefore not a reliable analytic variance normalization, although module size may still be used separately to allocate intervention sensitivity as in Section 3.3.

B.4 Location–Scale Calibration of the Online Score

The score equation (4) is the online counterpart of the ideal innovation equation (37): causal EMA estimates replace the unknown center and scale. We next characterize location–scale invariance under correspondingly scaled initial states, before relating this conditional result to the fixed initialization used in practice.

Proposition B.4 (location–scale invariance). Fix the decay coefficients $\alpha, \beta \in (0,1)$ in the alignment-statistic update equation (5). Only for this proposition, set the numerical stabilizer ε_{var} in the score equation (4) to zero. Within a stationary domain, suppose

$$C_l^t = \bar{C}_l + \sigma_{C,l}\epsilon_l^t, \qquad \sigma_{C,l} > 0, \tag{43}$$

where $\bar{C}_l$ and $\sigma_{C,l}$ are the tensor-specific within-domain alignment center and scale. The process $\{\epsilon_l^t\}_t$ is standardized, with $\mathbb{E}[\epsilon_l^t] = 0$ and $\operatorname{Var}(\epsilon_l^t) = 1$. Gaussianity and temporal independence are not required. At some t_0, suppose $E_l^{t_0} = \bar{C}_l + \sigma_{C,l}\widetilde{E}_l^{t_0}$ and $V_l^{t_0} = \sigma_{C,l}^2\widetilde{V}_l^{t_0}$ with $\widetilde{V}_l^{t_0} > 0$. Then, pathwise,

$$\begin{aligned} &E_l^t = \bar{C}_l + \sigma_{C,l}\widetilde{E}_l^t, \quad \widetilde{E}_l^t = \alpha\widetilde{E}_l^{t-1} + (1-\alpha)\epsilon_l^t, \\ &V_l^t = \sigma_{C,l}^2\widetilde{V}_l^t, \qquad\quad \widetilde{V}_l^t = \beta\widetilde{V}_l^{t-1} + \alpha(1-\beta)(\epsilon_l^t - \widetilde{E}_l^{t-1})^2, \\ &Z_l^t = \frac{\widetilde{E}_l^{t-1} - \epsilon_l^t}{\sqrt{\widetilde{V}_l^{t-1}}}. \end{aligned} \tag{44}$$

The first two identities hold for $t \geq t_0$, and the score identity holds for $t > t_0$. Hence the score does not depend on $\bar{C}_l$ or $\sigma_{C,l}$. If two tensors share the joint law of $(\widetilde{E}_l^{t_0}, \widetilde{V}_l^{t_0}, \{\epsilon_l^t\}_{t>t_0})$, their score distributions are identical at each $t > t_0$. Their score variances are therefore also identical whenever finite.

Proof. From the mean component of equation (5),

$$C_l^t - E_l^t = \alpha(C_l^t - E_l^{t-1}). \tag{45}$$

Substituting this identity into the variance component of equation (5) gives

$$V_l^t = \beta V_l^{t-1} + \alpha(1-\beta)(C_l^t - E_l^{t-1})^2. \tag{46}$$

The initialization in the proposition establishes the two state identities in equation (44) at t_0. Suppose they hold at $t-1$. Equation (43) then gives

$$C_l^t - E_l^{t-1} = \sigma_{C,l}(\epsilon_l^t - \widetilde{E}_l^{t-1}), \tag{47}$$

$$E_l^t = \alpha(\bar{C}_l + \sigma_{C,l}\widetilde{E}_l^{t-1}) + (1-\alpha)(\bar{C}_l + \sigma_{C,l}\epsilon_l^t) = \bar{C}_l + \sigma_{C,l}\widetilde{E}_l^t,$$

$$V_l^t = \beta\sigma_{C,l}^2\widetilde{V}_l^{t-1} + \alpha(1-\beta)\sigma_{C,l}^2(\epsilon_l^t - \widetilde{E}_l^{t-1})^2 = \sigma_{C,l}^2\widetilde{V}_l^t. \tag{48}$$

Thus the identities hold at t, completing the induction. The variance recursion also gives $\widetilde{V}_l^t \geq \beta \widetilde{V}_l^{t-1} > 0$. Finally, substituting the time-$(t-1)$ state and equation (43) into the score equation (4) yields

$$Z_l^t = \frac{\bar{C}_l + \sigma_{C,l}\widetilde{E}_l^{t-1} - (\bar{C}_l + \sigma_{C,l}\epsilon_l^t)}{\sqrt{\sigma_{C,l}^2 \widetilde{V}_l^{t-1}}} = \frac{\widetilde{E}_l^{t-1} - \epsilon_l^t}{\sqrt{\widetilde{V}_l^{t-1}}}, \tag{49}$$

which proves the claimed location–scale invariance. □

This is a conditional calibration result. Identical cross-tensor score laws require the same joint law of the standardized initial state and innovation process, as stated in the proposition. The implementation differs from the idealized scaling in two ways. First, it initializes every tensor with $V_l^{t_0} = \varepsilon_{\mathrm{var}}$, so the standardized initial value is

$$\widetilde{V}_l^{t_0} = \frac{\varepsilon_{\mathrm{var}}}{\sigma_{C,l}^2}, \tag{50}$$

which is scale dependent when $\sigma_{C,l}$ differs across tensors. This initialization contributes a term that decays geometrically with the variance recursion. Second, the fixed denominator term $\varepsilon_{\mathrm{var}}$ in equation (4) breaks exact positive-scale invariance. The practical score therefore approximates the proposition only after enough valid observations have made the initial-state contribution negligible and when $\sqrt{V_l^{t-1}} \gg \varepsilon_{\mathrm{var}}$. Arbitrary tensors may also differ in higher-order innovation shape. The proposition thus establishes an invariance property of the online recursion under scaled initialization, rather than universal finite-time calibration for arbitrary gradient processes.

B.5 Raw-Alignment Ablation

The raw-C^t variant compared in Section 3.2 removes the online standardization in equation (4). All entries in Table 1 use ImageNet-C with ResNet-50 and Adam, and both variants are evaluated on the same five seeds. For tensor l, the variant stores valid raw alignments C_l^τ in a FIFO window $\mathcal{W}_{C,l}^{t-1}$ available before the decision at step t, with $n_{C,l}^{t-1} = |\mathcal{W}_{C,l}^{t-1}|$. Replacing Z^τ by C_l^τ in equation (6) gives

$$\begin{cases} \widehat{\mu}_{C,l}^{t-1} = \dfrac{1}{n_{C,l}^{t-1}} \displaystyle\sum_{\tau \in \mathcal{W}_{C,l}^{t-1}} C_l^\tau, \\ (\widehat{\sigma}_{C,l}^{t-1})^2 = \dfrac{1}{n_{C,l}^{t-1} - 1} \displaystyle\sum_{\tau \in \mathcal{W}_{C,l}^{t-1}} (C_l^\tau - \widehat{\mu}_{C,l}^{t-1})^2. \end{cases}$$

After the same $W_{\min}$-sample warm-up used by P-DORM, the raw variant triggers when

$$C_l^t < \widehat{\mu}_{C,l}^{t-1} - k_l \widehat{\sigma}_{C,l}^{t-1}. \tag{51}$$

All other detector, threshold-allocation, and reset components remain unchanged. Thus, the ablation isolates whether standardizing C_l^t into Z_l^t improves reset decisions.

C Parameter-Aware Allocation-Shape Variants

This section details the five variants compared in the allocation-shape experiment of Section 4.3 and Table 5. In equation (10), $s(w_{m(l)})$ denotes the module-size term that replaces the default $2w_{m(l)} - 1$. The ablation keeps k_0, λ, clipping, and the tensor-wise mean calibration of Appendix A fixed, changing only $s(w)$:

$$\begin{aligned} &s_{\mathrm{fixed}}(w) = 0, && s_{\mathrm{small}}(w) = 2w - 1, && s_{\mathrm{large}}(w) = 1 - 2w, \\ &s_{\mathrm{mid}}(w) = 6(w - \tfrac{1}{2})^2 - \tfrac{1}{2}, && s_{\mathrm{ends}}(w) = \tfrac{1}{2} - 6(w - \tfrac{1}{2})^2. \end{aligned} \tag{52}$$

Because a smaller coefficient lowers the reset threshold, **Small** prioritizes smaller modules, whereas **Large** prioritizes larger ones. **Mid** prioritizes modules near the center of the size ranking, while **Ends** prioritizes the smallest and largest modules. **Fixed** removes size dependence. For every non-fixed shape, the calibration offset is recomputed to preserve the tensor-wise mean coefficient k_0. The variants therefore differ only in how reset sensitivity is distributed across modules. If all

monitored modules have equal size, equation (13) gives $w_m = 1/2$ throughout. Under the default Small allocation this makes $s_{\text{small}}(w_m) = 0$ and $k_l = k_0$, so no size heterogeneity is introduced. In Table 5(a), Small uses the five main-result seeds, whereas each alternative shape uses three seeds; each reported comparison retains the same data order and hyperparameters within its seed set.

D Detailed Online Procedure and Optimizer Extensions

Algorithm 2 Detailed P-DORM procedure with SGD

Require: Test stream $\{X^t\}_{t=1}^T$, model f_θ, SGD optimizer, FIFO capacity W, $W_{\min}$, (α, β), $\varepsilon_{\text{var}} = 10^{-4}$, and $\varepsilon_{\text{norm}} = 10^{-6}$

Compute and cache $\{k_l\}_{l=1}^L$ using equation (10) and Appendix A
for $l = 1, \ldots, L$ **do**
    $E_l \leftarrow 0$, $V_l \leftarrow \varepsilon_{\text{var}}$, and $N_l \leftarrow 0$
    $(\mathcal{W}_l, \mu_l, \sigma_l) \leftarrow (\varnothing, 0, 0)$
end for
for $t = 1, \ldots, T$ **do**
    Predict on X^t and compute the full gradient g^t using equation (2)
    for $l = 1, \ldots, L$ **do**
        if g_l^t is absent **then**
            continue
        end if
        Read g_l^t and lazily initialize a missing u_l^{t-1} to zero
        if $t > 1$ **then**
            Compute C_l^t by equation (3) with norm floor $\varepsilon_{\text{norm}}$
            $a_l^t \leftarrow \mathbf{1}[\|g_l^t\|_2 \geq \varepsilon_{\text{norm}} \wedge \|u_l^{t-1}\|_2 \geq \varepsilon_{\text{norm}}]$
            $r_l^t \leftarrow 0$
            if $a_l^t = 1 \wedge N_l > 1$ **then**
                Compute the tensor-indexed Z_l^t from (E_l, V_l) using equation (4)
                Compute r_l^t using equation (8)
                if $r_l^t = 1$ **then**
                    $u_l^{t-1} \leftarrow 0$ using equation (11)
                    $(\mathcal{W}_l, \mu_l, \sigma_l) \leftarrow (\varnothing, 0, 0)$
                else
                    Append Z_l^t to $\mathcal{W}_l$ and retain the latest W entries
                    if $|\mathcal{W}_l| = 1$ **then**
                        $(\mu_l, \sigma_l) \leftarrow (Z_l^t, 0)$
                    else
                        Recompute (μ_l, σ_l) using equation (6)
                    end if
                end if
            end if
            if $N_l = 0$ **then**
                $E_l \leftarrow C_l^t$ and $V_l \leftarrow \varepsilon_{\text{var}}$
            else
                Update the tensor-indexed (E_l, V_l) using equation (5)
            end if
            $N_l \leftarrow N_l + 1$
        end if
        Apply the original SGD update to (θ_l, u_l) using g_l^t
    end for
end for

The $|\mathcal{W}_l| = 1$ branch prevents the sample-variance calculation from using a zero denominator. This fallback cannot produce a reset because equation (8) is evaluated only after $|\mathcal{W}_l| \geq W_{\min} = 15$ valid scores have accumulated. The first post-initialization alignment initializes (E_l, V_l), and the second updates these statistics. Scoring begins from the third alignment so that Z_l^t uses a pre-decision variance informed by a genuine residual rather than only the fixed initialization.

State-transition details. A triggering alignment updates (E_l, V_l) after the decision but is not appended to the newly cleared window. The current gradient is then incorporated exactly once by the original optimizer update. If a parameter has no gradient, its optimizer and detector states are unchanged. A numerically zero gradient is different: flooring the denominator norms at $\varepsilon_{\mathrm{norm}}$ keeps the causal statistic recursion finite, while the score, window update, and reset decision are gated and the original optimizer step still advances. Optimizer buffers are created lazily per parameter, and a newly created zero buffer cannot produce a valid anomaly score or reset decision until it has acquired a nonzero historical direction. When an underlying method such as RDumb restores a saved optimizer state, the detector variables $(E_l, V_l, N_l, \mathcal{W}_l, \mu_l, \sigma_l)$ stored in that state are restored synchronously. Restoring the initial empty optimizer snapshot therefore restarts both the optimizer and detector states on the next update.

D.1 Extension to Adam

For Adam, let m_l and v_l denote the first and second moments of tensor l. In equation (3), P-DORM replaces u_l^{t-1} with the pre-update first moment m_l^{t-1}; the detector is otherwise unchanged. When tensor l triggers, P-DORM preserves θ_l and applies

$$m_l \leftarrow 0, \qquad v_l \leftarrow \tau_v v_l, \tag{53}$$

where $\tau_v \in [0, 1]$ controls retention of the squared-gradient scale. For bias correction, the first-moment decay product q_l restarts from 1, whereas the second-moment count t_l is retained. The triggering gradient is then incorporated once, after which the corrections use $1 - q_l$ and $1 - \beta_2^{t_l}$, respectively. Missing gradients advance neither quantity. Without a trigger, $q_l = \beta_1^{t_l}$ and the update reduces exactly to Pure Adam. The extension is evaluated in Tables 2–4, and Appendix G.2 analyzes τ_v under this fixed rule.

D.2 Extensions to Other Optimizers

The same detector and parameter-aware thresholds apply after identifying each optimizer's directional state. For uncentered RMSProp with momentum (PyTorch Contributors, 2026), equation (3) compares the pre-update momentum buffer b_l^{t-1} with the provisional preconditioned gradient computed from g_l^t and the pre-update squared-gradient state v_l^{t-1}. The provisional state is not committed. After a trigger, the original RMSProp recurrence is recomputed from the reset state, so the current gradient enters each state exactly once. For AdaBelief (Zhuang et al., 2020) and Lion (Chen et al., 2023), P-DORM replaces u_l^{t-1} in equation (3) with the corresponding pre-update first moment m_l^{t-1}.

When tensor l triggers, P-DORM preserves θ_l and applies

$$\begin{aligned} \text{RMSProp:} \quad & (b_l, v_l) \leftarrow (0, \tau_v v_l), \\ \text{AdaBelief:} \quad & (m_l, s_l, q_l) \leftarrow (0, \tau_v s_l, 1), \\ \text{Lion:} \quad & m_l \leftarrow 0. \end{aligned} \tag{54}$$

Here, v_l is the RMSProp squared-gradient state, s_l is the AdaBelief belief-variance state, and q_l is the cumulative first-moment decay factor used for local bias correction. After an AdaBelief reset, the current gradient first updates m_l, and the belief variance then uses the same-step residual $g_l^t - m_l^t$, matching the original update order; the gradient is accumulated only once. Lion has no second-order scale state. Missing gradients skip all optimizer-specific counters and states. Figure 1 evaluates these extensions.

E Datasets and Evaluation Protocols

E.1 Datasets

The three corruption benchmarks each contain 15 corruption types at five severity levels. DomainNet-126 instead contains four natural visual domains, with Real used as the source domain and Clipart, Painting, and Sketch used as target domains. The complete domain composition is specified below.

CIFAR-10-C (C10C). CIFAR-10-C applies 15 common corruptions at five severity levels to the 10,000-image CIFAR-10 test set (Krizhevsky, 2009; Hendrycks & Dietterich, 2019). Each corruption and severity pair defines a target domain. CIFAR-10-C retains the 10 classes and 32×32 resolution of CIFAR-10. The corruption domains comprise Gaussian, shot, and impulse noise, defocus, glass, motion, and zoom blur, as well as snow, frost, fog, brightness, contrast, elastic transform, pixelation, and JPEG compression. We use WideResNet-28-10 as the source model (Zagoruyko & Komodakis, 2016).

CIFAR-100-C (C100C). CIFAR-100-C applies the same 15 corruption domains and five severity levels to the 10,000-image CIFAR-100 test set and retains the 100 classes and 32×32 resolution of CIFAR-100 (Krizhevsky, 2009; Hendrycks & Dietterich, 2019). Each corruption and severity pair again defines a target domain. We use an AugMix-pretrained ResNeXt-29 as the source model (Xie et al., 2017; Hendrycks et al., 2020).

ImageNet-C (IN-C). ImageNet-C applies the same 15 corruption domains and five severity levels to the 50,000-image, 1,000-class ImageNet validation set (Deng et al., 2009; Hendrycks & Dietterich, 2019). Each corruption and severity pair defines a target domain. We evaluate ResNet-50 and ViT-B/16 source models (He et al., 2016; Dosovitskiy et al., 2021). All paired variants use identical samples.

DomainNet-126 (DN-126). DomainNet is a natural multi-domain recognition benchmark whose domains correspond to distinct visual styles rather than synthetic corruptions (Peng et al., 2019). Following prior TTA evaluation, we use the 126-class subset with Real, Clipart, Painting, and Sketch (Döbler et al., 2023). Real is the source domain with 20,939 listed images. Clipart, Painting, and Sketch are the three target domains with 5,611, 9,081, and 7,313 images, respectively, and adaptation follows this order. We use ResNet-50 in the main evaluation and additionally report ViT-B/16 in Appendix G.1 (He et al., 2016; Dosovitskiy et al., 2021). Because DN-126 has no ordered severity axis, GTTA is not evaluated on this dataset.

E.2 Evaluation Protocols

All protocols perform one-pass, online, source-free adaptation. Ground-truth labels are never exposed to the adaptation algorithm. Labels are used only offline to construct PTTA streams and after prediction to compute accuracy. Following the benchmark configurations used in prior TTA studies (Wang et al., 2021; Döbler et al., 2023), we use batch size 200 on C10C and C100C and 64 on IN-C and DN-126. Unless a protocol specifies reinitialization, adapted parameters, optimizer states, and P-DORM detector statistics persist throughout the stream. Paired methods share the same samples, ordering, and random seeds. Following standard corruption and continual-adaptation evaluations (Hendrycks & Dietterich, 2019; Wang et al., 2022; Döbler et al., 2023), the default corruption severity is 5.

TTA. Following the standard TTA protocol (Wang et al., 2021), the model adapts to one target domain at a time. Before each corruption or natural target domain, the model parameters, optimizer state, and P-DORM detector statistics are reinitialized from the same source state. TTA therefore isolates within-domain adaptation and prevents information from propagating across target domains.

CTTA (continual test-time adaptation). CTTA presents target domains sequentially without revealing or resetting at their boundaries (Wang et al., 2022; Döbler et al., 2023). For each corruption dataset, the stream contains 15 severity-5 domains in the corruption order listed above, producing 14 boundaries between adjacent corruption blocks. For DN-126, the three target domains follow Clipart $\rightarrow$ Painting $\rightarrow$ Sketch, producing two domain boundaries. This protocol evaluates adaptation under abrupt distribution shifts together with resistance to drift and forgetting.

GTTA (gradual test-time adaptation). Following the gradual-shift setting of Döbler et al. (2023), we construct GTTA by replacing each abrupt corruption block with the severity trajectory

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1,$$

without resetting between adjacent severities or corruptions. This protocol evaluates incremental drift and is reported only for C10C, C100C, and IN-C.

PTTA (practical test-time adaptation). PTTA combines sequential domain shifts with temporally correlated class order and time-varying label marginals (Gong et al., 2022; Yuan et al., 2023). Within each target domain, samples from each class are allocated among temporally ordered chunks using a symmetric Dirichlet distribution. A smaller concentration α yields stronger class correlation. We use $\alpha = 0.3$ for C10C to avoid nearly deterministic class blocks in the 10-class stream and $\alpha = 0.1$ for C100C, IN-C, and DN-126 to impose strong correlation in their larger label spaces. Domain blocks remain sequential, without resets between them.

Mixed-domain TTA (Mixed). Following mixed-domain evaluation in universal TTA (Marsden et al., 2024), samples from all 15 severity-5 corruptions, or all three DN-126 target domains, are pooled and randomly interleaved into one stream. The adaptation method receives no domain identifier and is never reset. The pooled distribution is fixed, but the domain composition of consecutive mini-batches can vary, making this protocol a robustness test under mixed-domain inputs rather than an ordered domain-shift sequence.

DomainNet-126 aggregation. For TTA, CTTA, and PTTA, DN-126 error is the unweighted macro-average of the three separately evaluated target-domain errors. Mixed instead evaluates the pooled stream and reports the sample-weighted micro-average:

$$\mathrm{Err}_{\mathrm{macro}} = \frac{1}{3}\sum_{d=1}^{3} \mathrm{Err}_d, \qquad \mathrm{Err}_{\mathrm{micro}} = \frac{\sum_{d=1}^{3} n_d \,\mathrm{Err}_d}{\sum_{d=1}^{3} n_d}. \tag{55}$$

Because Clipart, Painting, and Sketch have different sizes, these summaries differ even for the same fixed source model and samples. This explains the Source-only changes from 45.28 to 44.67 in Table 4 and from 31.46 to 30.94 in Table 6; they do not reflect a change in model, preprocessing, or evaluated examples.

Relationship among the protocols. The five protocols vary whether adaptation is reinitialized at known boundaries, whether domains arrive in blocks or are interleaved, and whether shifts are abrupt, gradual, or coupled with label correlation. **TTA** reinitializes at known boundaries. **CTTA** retains abrupt domain blocks without resets, **GTTA** smooths severity transitions, and **PTTA** adds temporal label correlation. **Mixed** removes stable domain blocks altogether.

F Implementation Details

F.1 Main Experiments

Evaluation and hyperparameters. Main results report averages over five seeds. For every method, dataset, optimizer, and test-stream setting, Pure and P-DORM use the same frozen learning rate. Unless otherwise specified, P-DORM uses $k_0 = 2.5$, $k_{\min} = k_0 - 0.5$, $k_{\max} = k_0 + 0.5$, $\lambda = 0.5$, $\alpha = \beta = 0.9$, $W = 100$, $W_{\min} = 15$, $\varepsilon_{\mathrm{var}} = 10^{-4}$, and $\varepsilon_{\mathrm{norm}} = 10^{-6}$. The Adam reset retains half of the second moment with $\tau_v = 0.5$.

Optimizer configurations. Main experiments use Adam with $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$, and SGD with momentum 0.98, Nesterov updates, and zero dampening. Both use zero weight decay and one optimizer update per batch. For the three additional optimizers in Figure 1, RMSProp with momentum uses momentum 0.9, squared-gradient decay 0.99, $\epsilon = 10^{-8}$, and no centering. AdaBelief uses $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-16}$. Lion uses $(\beta_1, \beta_2) = (0.9, 0.99)$. These three also use zero weight decay.

Architecture-specific adapted parameters. The implementations of TENT, DeYO, EATA-C, and CoME collect the affine scale and bias from supported normalization layers, while RDumb inherits the same parameter-selection rule from TENT (Wang et al., 2021; Lee et al., 2024; Tan et al., 2025; Zhang et al., 2025; Press et al., 2023). BufferTTA additionally applies its method-specific buffer mechanism (Kim et al., 2025). Consequently, experiments with ViT-B/16 adapt LayerNorm affine parameters (Ba et al., 2016), whereas experiments with ResNet-based backbones adapt BatchNorm affine parameters and any method-specific normalization state (Ioffe & Szegedy, 2015). P-DORM operates on the optimizer states associated with whichever parameters the underlying TTA method already updates and does not expand the trainable parameter set.

Boundary-Oracle intervention. Boundary-Oracle receives the known domain transitions and applies the optimizer-specific reset to every initialized parameter state while preserving model parameters. It clears SGD momentum, and for Adam, RMSProp, and AdaBelief it clears the directional state and applies $v_l \leftarrow \tau_v v_l$ or $s_l \leftarrow \tau_v s_l$ to the scale state; Lion clears its first moment. The plain oracle optimizers retain their native global counters and bias-correction rules. Boundary-Oracle therefore differs from P-DORM mainly in trigger availability and tensor coverage, while P-DORM's Adam and AdaBelief implementations additionally use the local first-moment correction described in Appendix D.1.

F.2 Training Setting of ViT-B/16 on DomainNet-126

Because no directly compatible ViT-B/16 source checkpoint is available for DomainNet-126, we train one for this evaluation. We initialize ViT-B/16 (Dosovitskiy et al., 2021) from ImageNet-21K-pretrained and ImageNet-1K-fine-tuned AugReg weights and replace its classifier with a 126-class linear head. The 20,939 Real-domain images from DomainNet-126 (Peng et al., 2019) are split class-wise into 19,892 training and 1,047 validation images using seed 2020. We fine-tune all parameters for at most 20 epochs with AdamW (Loshchilov & Hutter, 2019), batch size 64, learning rate 3×10^{-5}, weight decay 0.05, two warm-up epochs followed by cosine decay, and cross-entropy with label smoothing 0.1. The checkpoint is selected solely by Real-domain validation accuracy.

TTA evaluation uses the complete Clipart, Painting, and Sketch streams under the TTA, CTTA, PTTA, and Mixed settings. PTTA uses Dirichlet concentration 0.1, and results are averaged over seeds $\{1, 2, 3, 7, 14\}$. The evaluated methods adapt LayerNorm affine parameters, and each Pure/P-DORM pair uses the same frozen learning rate.

F.3 Supervised Non-Stationary Training Configuration

We start from the Real-domain ResNet-50 checkpoint released with AdaContrast (Chen et al., 2022). Clipart (C), Painting (P), and Sketch (S) are split class-wise into 17,604 training and 4,401 validation images, and each training split is divided into three disjoint shards. The seed-1 training stream is

$$C_1 \to S_1 \to P_1 \to C_2 \to S_2 \to P_2 \to C_3 \to S_3 \to P_3.$$

Seeds 1, 2, 3, 7, and 14 use the base orders $C \to S \to P$, $P \to C \to S$, $P \to S \to C$, $S \to C \to P$, and $S \to P \to C$, respectively, each repeated for three cycles as above. The k-th visit to a domain uses its k-th shard, so every training image appears once and the stream contains eight abrupt domain transitions. All methods share the same splits, order, batches, and within-shard shuffling.

All model parameters are optimized with cross-entropy loss, while BatchNorm running statistics remain frozen. We use batch size 64, full precision, a constant learning rate of 3×10^{-4}, SGD momentum $\beta = 0.98$, and weight decay 10^{-4}. Pure carries momentum throughout the stream, Boundary-Oracle clears it at the eight known boundaries, and P-DORM receives no boundary information. If d_j is the domain trained in stage j, stage error evaluates that same domain immediately after the stage and averages the nine results. Final error evaluates all three target domains after stage 9 and macro-averages them:

$$\mathrm{Err}_{\mathrm{stage}} = \frac{1}{9}\sum_{j=1}^{9} \mathrm{Err}_{d_j}^{(j)}, \qquad \mathrm{Err}_{\mathrm{final}} = \frac{1}{3}\sum_{d\in\{C,P,S\}} \mathrm{Err}_{d}^{(9)}. \tag{56}$$

G Additional Experiments

G.1 DomainNet-126 with ViT-B/16

We additionally evaluate a ViT-B/16 source model on DomainNet-126. Appendix F.2 describes the source-model training and TTA configuration. Table 6 shows that P-DORM lowers the across-method error in every reported setting for both Adam and SGD.

Using paired seed-level differences and first averaging across the 24 method–setting cells for each optimizer, the across-method changes are -0.07 ± 0.01 points for Adam and -0.07 ± 0.03 points for

Table 6: **Additional results on DomainNet-126 with ViT-B/16.** Five-seed mean error (%). GTTA is omitted because DN-126 has no ordered severity axis for constructing gradual transitions. Notation and highlighting follow Table 2.

		Adam					SGD				
Method	Variant	TTA	CTTA	PTTA	Mix	Avg.	TTA	CTTA	PTTA	Mix	Avg.
Source-only	–	31.46	31.46	31.46	30.94	31.33	31.46	31.46	31.46	30.94	31.33
TENT (Wang et al., 2021)	Pure	31.41	31.45	31.40	30.93	31.30	31.67	31.69	31.88	30.96	31.55
	P-DORM	31.40	31.45	31.42	30.94	31.30	31.53	31.50	31.58	30.96	31.39
	Δ	**-0.01**	0.00	+0.02	+0.01	+0.01	**-0.14**	**-0.19**	**-0.30**	0.00	**-0.16**
DeYO (Lee et al., 2024)	Pure	31.88	31.73	31.74	30.95	31.57	31.58	31.55	31.49	30.97	31.40
	P-DORM	31.86	31.41	31.34	30.95	31.39	31.57	31.50	31.48	30.97	31.38
	Δ	**-0.02**	**-0.32**	**-0.40**	0.00	**-0.19**	**-0.01**	**-0.05**	**-0.01**	0.00	**-0.02**
CoME (Zhang et al., 2025)	Pure	29.73	29.05	29.10	28.57	29.11	29.44	29.24	29.17	28.48	29.08
	P-DORM	29.73	28.98	29.02	28.56	29.07	29.29	29.17	29.17	28.55	29.05
	Δ	0.00	**-0.07**	**-0.08**	**-0.01**	**-0.04**	**-0.15**	**-0.07**	0.00	+0.07	**-0.04**
BufferTTA (Kim et al., 2025)	Pure	31.43	31.45	31.43	30.95	31.31	31.46	31.68	31.48	31.21	31.46
	P-DORM	31.43	31.45	31.43	30.94	31.31	31.46	31.45	31.45	30.95	31.33
	Δ	0.00	0.00	0.00	**-0.01**	0.00	0.00	**-0.23**	**-0.03**	**-0.26**	**-0.13**
EATA-C (Tan et al., 2025)	Pure	31.04	31.41	31.53	30.86	31.21	31.47	31.89	31.89	31.27	31.63
	P-DORM	31.03	31.39	31.43	30.87	31.18	31.44	31.58	31.56	31.02	31.40
	Δ	**-0.01**	**-0.02**	**-0.10**	+0.01	**-0.03**	**-0.03**	**-0.31**	**-0.33**	**-0.25**	**-0.23**
RDumb (Press et al., 2023)	Pure	31.29	31.30	31.33	30.60	31.13	31.38	30.91	30.82	30.20	30.83
	P-DORM	31.27	31.07	31.15	30.48	30.99	31.48	31.11	30.88	30.33	30.95
	Δ	**-0.02**	**-0.23**	**-0.18**	**-0.12**	**-0.14**	+0.10	+0.20	+0.06	+0.13	+0.12
Across methods	Avg. Δ	**-0.01**	**-0.11**	**-0.12**	**-0.02**	**-0.07**	**-0.04**	**-0.11**	**-0.10**	**-0.05**	**-0.07**

SGD (mean ± standard deviation over five seeds). The effect remains method dependent: RDumb with SGD increases its average error by 0.12 points, whereas the across-method SGD average decreases by 0.07 points.

G.2 Sensitivity to Adam Second-Moment Retention

For Adam, P-DORM removes the stale first-moment direction after a trigger and softly retains the second-moment scale through $v_l \leftarrow \tau_v v_l$, where $\tau_v \in [0, 1]$. Appendix D.1 gives the complete reset rule. The endpoints respectively discard and fully retain the accumulated scale information. Figure 3 evaluates this trade-off across representative methods, datasets, and non-stationary test streams.

The response is configuration dependent. Complete removal ($\tau_v = 0$) is detrimental for TENT and DeYO on CIFAR-10-C under the CTTA setting, while stronger retention increases error for TENT on ImageNet-C and DeYO on CIFAR-100-C in the displayed GTTA settings. Importantly, $\tau_v = 0$ is not equivalent to restarting a fresh Adam instance: P-DORM retains the second-moment count t_l while restarting the first-moment decay product q_l. The sensitivity results therefore use the fixed post-reset bias-correction rule described in Appendix D.1. Performance at $\tau_v = 0$ can reflect both removal of the accumulated scale state and this correction convention, so it should not be attributed solely to loss of historical scale information. Other configurations are comparatively insensitive over a broad range, and partial retention provides a practical compromise.

G.3 Momentum-Free Baseline

We test whether removing momentum can replace optimizer-state management. Momentum-free SGD disables momentum, whereas Pure and P-DORM follow the main-experiment configuration. All variants use the same frozen learning rate within each method and setting. Because resetting the momentum buffer is a no-op without momentum, we report one momentum-free arm. Table 7 summarizes the comparison.

Momentum-free SGD has the highest error in 16 of 18 configurations, including all 12 CTTA and GTTA comparisons. Its two non-worst cases occur on CIFAR-10-C under PTTA. P-DORM attains the lowest error in 15 of 18 configurations. Under this matched-LR protocol, removing momentum does not reproduce the benefit of P-DORM, which instead retains useful accumulated history and resets it only when it becomes incompatible with the current gradient.

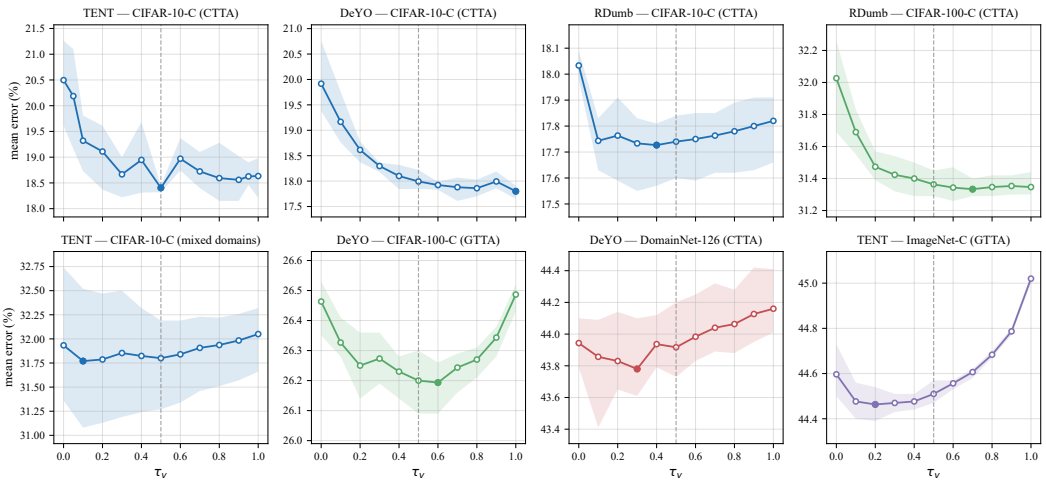


Figure 3: **Sensitivity to Adam second-moment retention τ_v.** Mean error rate (%) over three seeds. CIFAR-10-C uses WideResNet-28-10, CIFAR-100-C uses ResNeXt-29, and ImageNet-C uses ResNet-50. Shaded bands denote one standard deviation. A smaller τ_v forgets more accumulated scale information after a reset, while a larger value retains more. The dashed line marks $\tau_v = 0.5$. Lower is better.

Table 7: **Momentum-free SGD baseline.** Five-seed mean error (%). Pure and P-DORM follow the main-experiment configuration, and learning rates are matched within each configuration. Red denotes the highest error and bold the lowest.

Method	Variant	CIFAR-10-C			CIFAR-100-C			ImageNet-C		
		CTTA	GTTA	PTTA	CTTA	GTTA	PTTA	CTTA	GTTA	PTTA
TENT (Wang et al., 2021)	Momentum-free	19.95	12.89	**66.04**	34.75	28.67	43.04	67.78	46.57	68.29
	Pure	18.39	12.28	67.74	33.20	27.95	**41.96**	63.48	45.16	64.36
	P-DORM	**18.15**	**12.14**	66.89	**32.87**	**27.86**	42.02	**63.27**	**44.92**	**64.11**
DeYO (Lee et al., 2024)	Momentum-free	19.99	12.90	66.04	34.45	27.55	42.91	68.04	46.17	68.54
	Pure	17.85	12.31	66.34	32.62	27.38	41.60	64.05	**43.84**	65.29
	P-DORM	**17.71**	**12.11**	**65.99**	**32.14**	**27.22**	**41.39**	**63.91**	43.91	**64.74**

G.4 Seed-Wise Variability

Tables 8–10 report sample standard deviations computed with the $n - 1$ denominator over seeds $\{1, 2, 3, 7, 14\}$. The Pure and P-DORM rows quantify variation in each arm, whereas the Δ rows use paired seed-wise differences and therefore directly characterize uncertainty in the reported comparison. Source-only is omitted because it is evaluated once.

G.5 Computational Efficiency

We profile Pure and P-DORM on a single NVIDIA RTX 4090 using ResNet-50, ViT-B/16, and WideResNet-28-10. Each measurement uses 120 warm-up steps followed by 300 timed steps and is averaged over three repetitions and five optimizers. Batch size is 64 for ResNet-50 and ViT-B/16 and 200 for WideResNet-28-10. Table 11 reports the additional cost of the P-DORM implementation relative to Pure.

P-DORM introduces no trainable parameters and reuses the historical direction already maintained by the optimizer. Its additional GPU-resident detector state consists of two 0-dimensional scalars per monitored tensor, namely the alignment statistics E and V, producing a 0.6–1.3 KB increase. The remaining scalar bookkeeping and the FIFO window $\mathcal{W}^t$ reside in CPU memory. Table 11 includes the FIFO's compact float32 payload of 66, 61, and 32 KB for ResNet-50, ViT-B/16, and WideResNet-28-10. The complete Python deque representation, including object overhead, remains

Table 8: **Seed-wise variability on CIFAR-10-C and CIFAR-100-C.** Sample standard deviation of error (%) over five seeds. Pure and P-DORM rows report per-arm deviations; Δ rows report the deviation of paired seed differences. Avg. is the arithmetic mean of the displayed deviations, not a pooled estimate.

	Method	Variant	Adam						SGD					
			TTA	CTTA	GTTA	PTTA	Mix	Avg.	TTA	CTTA	GTTA	PTTA	Mix	Avg.
(a) CIFAR-10-C	TENT (Wang et al., 2021)	Pure	0.10	0.33	0.06	0.58	0.35	0.29	0.08	0.32	0.23	0.66	0.88	0.44
		P-DORM	0.11	0.21	0.07	0.59	0.52	0.30	0.08	0.08	0.08	0.57	0.62	0.29
		Δ	0.01	0.15	0.03	0.01	0.19	0.08	0.01	0.27	0.21	0.31	0.78	0.32
	DeYO (Lee et al., 2024)	Pure	0.10	0.14	0.04	0.61	0.48	0.27	0.08	0.13	0.11	0.56	0.69	0.31
		P-DORM	0.10	0.17	0.03	0.60	0.41	0.26	0.08	0.22	0.13	0.57	0.40	0.28
		Δ	0.01	0.13	0.02	0.02	0.17	0.07	0.01	0.14	0.15	0.10	0.88	0.26
	CoME (Zhang et al., 2025)	Pure	0.19	0.06	0.05	0.58	0.07	0.19	0.08	0.15	0.07	0.34	1.12	0.35
		P-DORM	0.19	0.09	0.06	0.57	0.10	0.20	0.08	0.11	0.05	0.41	1.02	0.34
		Δ	0.01	0.04	0.01	0.04	0.04	0.03	0.01	0.15	0.07	0.40	0.48	0.22
	BufferTTA (Kim et al., 2025)	Pure	0.07	0.06	0.08	0.59	0.03	0.16	0.08	0.51	0.30	0.74	1.03	0.53
		P-DORM	0.07	0.08	0.12	0.58	0.04	0.18	0.08	0.28	0.11	0.61	0.15	0.25
		Δ	0.01	0.04	0.07	0.01	0.01	0.03	0.00	0.45	0.30	0.16	1.10	0.40
	EATA-C (Tan et al., 2025)	Pure	0.07	0.04	0.05	0.62	0.16	0.19	0.09	0.08	0.14	0.79	0.43	0.31
		P-DORM	0.07	0.06	0.06	0.63	0.20	0.20	0.09	0.11	0.12	0.63	0.61	0.31
		Δ	0.01	0.05	0.04	0.03	0.08	0.04	0.01	0.06	0.08	0.17	0.29	0.12
	RDumb (Press et al., 2023)	Pure	0.08	0.11	0.06	0.60	0.19	0.21	0.11	0.15	0.05	0.48	0.37	0.23
		P-DORM	0.08	0.11	0.06	0.60	0.34	0.24	0.11	0.13	0.05	0.59	0.44	0.26
		Δ	0.01	0.09	0.01	0.01	0.23	0.07	0.02	0.03	0.03	0.13	0.19	0.08
	Across methods	Avg. $\mathrm{Std}(\Delta)$	0.01	0.08	0.03	0.02	0.12	0.05	0.01	0.18	0.14	0.21	0.62	0.23
(b) CIFAR-100-C	TENT (Wang et al., 2021)	Pure	0.10	0.06	0.07	0.35	0.05	0.12	0.05	0.10	0.15	0.41	0.30	0.20
		P-DORM	0.09	0.09	0.04	0.36	0.03	0.12	0.05	0.20	0.19	0.32	0.29	0.21
		Δ	0.02	0.09	0.04	0.03	0.05	0.05	0.01	0.12	0.05	0.12	0.21	0.10
	DeYO (Lee et al., 2024)	Pure	0.10	0.05	0.05	0.33	0.12	0.13	0.07	0.08	0.09	0.39	0.95	0.32
		P-DORM	0.09	0.05	0.08	0.33	0.14	0.14	0.07	0.08	0.08	0.30	0.71	0.25
		Δ	0.02	0.03	0.04	0.08	0.07	0.05	0.01	0.07	0.10	0.12	0.50	0.16
	CoME (Zhang et al., 2025)	Pure	0.11	0.07	0.04	0.35	0.19	0.15	0.06	0.10	0.09	0.38	0.35	0.20
		P-DORM	0.11	0.06	0.05	0.35	0.31	0.18	0.06	0.16	0.15	0.29	0.36	0.20
		Δ	0.01	0.02	0.04	0.06	0.14	0.05	0.00	0.13	0.09	0.14	0.25	0.12
	BufferTTA (Kim et al., 2025)	Pure	0.10	0.08	0.03	0.35	0.07	0.13	0.03	0.03	0.06	0.35	0.06	0.11
		P-DORM	0.09	0.11	0.05	0.35	0.07	0.13	0.03	0.03	0.03	0.35	0.06	0.10
		Δ	0.02	0.08	0.05	0.01	0.01	0.03	0.00	0.01	0.08	0.01	0.01	0.02
	EATA-C (Tan et al., 2025)	Pure	0.07	0.07	0.04	0.27	0.11	0.11	0.05	0.04	0.03	0.28	0.70	0.22
		P-DORM	0.07	0.08	0.08	0.30	0.08	0.12	0.04	0.16	0.05	0.29	0.51	0.21
		Δ	0.01	0.01	0.05	0.05	0.06	0.04	0.01	0.17	0.05	0.04	0.48	0.15
	RDumb (Press et al., 2023)	Pure	0.11	0.10	0.07	0.33	0.05	0.13	0.07	0.11	0.08	0.38	0.43	0.21
		P-DORM	0.10	0.08	0.07	0.33	0.07	0.13	0.09	0.20	0.09	0.36	0.38	0.22
		Δ	0.02	0.04	0.01	0.02	0.03	0.02	0.03	0.14	0.03	0.06	0.26	0.10
	Across methods	Avg. $\mathrm{Std}(\Delta)$	0.02	0.05	0.04	0.04	0.06	0.04	0.01	0.11	0.07	0.08	0.29	0.11

below 0.6 MB in these models. Its append and eviction costs are included in the measured latency. Construction also takes less than 0.6 ms. The total memory overhead is therefore negligible.

The current reference implementation does incur measurable latency. Profiling indicates that the overhead scales more closely with the number of monitored tensors than with trigger frequency. This behavior is consistent with repeated small GPU operations in the tensor-wise cosine similarity, norm reductions, and online-statistic updates, rather than with the inexpensive state-clearing operation itself. P-DORM therefore trades negligible memory for an 11.7–39.1% increase in per-step wall-clock time in the tested configurations. Batched statistics or fused kernels could reduce this implementation overhead without changing the method. These measurements distinguish the negligible parameter and memory overhead from the measurable latency cost.

H Scope and Methodological Clarifications

H.1 Interpretation of the Detection Signal

P-DORM is designed for non-stationary TTA streams, where distribution shifts can produce sustained changes in adaptation dynamics (Liang et al., 2025). An unusually large departure between the current gradient and optimizer state is therefore used as evidence that optimizer history may have become stale (Sahu et al., 2026). The score is an operational signal of optimizer-history incompatibility rather than a general-purpose or causal change-point detector. Similar departures can arise from local curvature or conflicting optimization dynamics without a distribution shift (O'Donoghue & Candès, 2015; Liang et al., 2026). P-DORM reduces sensitivity to ordinary fluctuations through tensor-specific standardization and window-based thresholding, but it neither identifies the underlying domain nor attributes a unique cause to a departure.

Table 9: **Seed-wise variability on ImageNet-C.** Definitions and notation follow Table 8.

	Method	Variant	Adam						SGD					
			TTA	CTTA	GTTA	PTTA	Mix	Avg.	TTA	CTTA	GTTA	PTTA	Mix	Avg.
(a) ResNet-50	TENT (Wang et al., 2021)	Pure	0.07	0.07	0.03	0.12	0.05	0.07	0.08	0.07	0.11	0.06	0.28	0.12
		P-DORM	0.07	0.10	0.05	0.13	0.06	0.08	0.09	0.07	0.12	0.13	0.21	0.13
		Δ	0.01	0.08	0.03	0.05	0.02	0.04	0.02	0.08	0.07	0.16	0.14	0.09
	DeYO (Lee et al., 2024)	Pure	0.08	0.18	0.08	0.11	0.16	0.12	0.10	0.21	0.09	0.57	2.04	0.60
		P-DORM	0.08	0.18	0.35	0.13	0.17	0.18	0.08	0.15	0.08	0.33	1.21	0.37
		Δ	0.01	0.10	0.29	0.07	0.07	0.11	0.04	0.20	0.09	0.47	1.38	0.44
	CoME (Zhang et al., 2025)	Pure	0.07	0.09	0.02	0.06	0.05	0.06	0.09	0.09	0.57	0.14	0.27	0.23
		P-DORM	0.07	0.10	0.05	0.09	0.06	0.07	0.10	0.13	0.20	0.06	0.14	0.13
		Δ	0.01	0.04	0.03	0.04	0.04	0.03	0.02	0.13	0.47	0.19	0.17	0.20
	BufferTTA (Kim et al., 2025)	Pure	0.05	0.06	0.04	0.24	0.18	0.12	0.07	0.06	0.10	0.21	0.47	0.18
		P-DORM	0.05	0.21	0.05	0.10	0.16	0.11	0.07	0.14	0.10	0.12	0.20	0.12
		Δ	0.03	0.18	0.02	0.22	0.23	0.14	0.01	0.14	0.04	0.21	0.28	0.14
	EATA-C (Tan et al., 2025)	Pure	0.06	0.09	0.13	0.10	0.24	0.12	0.07	0.17	0.91	0.23	0.84	0.45
		P-DORM	0.06	0.10	0.17	0.07	0.30	0.14	0.05	0.04	0.69	0.23	0.55	0.31
		Δ	0.00	0.04	0.16	0.04	0.13	0.07	0.02	0.17	0.31	0.08	0.81	0.28
	RDumb (Press et al., 2023)	Pure	0.10	0.18	0.11	0.07	0.47	0.19	0.24	0.28	0.16	0.38	0.18	0.25
		P-DORM	0.10	0.19	0.11	0.09	0.58	0.21	0.31	0.24	0.16	0.36	0.20	0.25
		Δ	0.02	0.09	0.03	0.08	0.27	0.10	0.09	0.12	0.03	0.10	0.04	0.08
	Across methods	Avg. $\mathrm{Std}(\Delta)$	0.01	0.09	0.09	0.08	0.13	0.08	0.03	0.14	0.17	0.20	0.47	0.20
(b) ViT-B/16	TENT (Wang et al., 2021)	Pure	0.39	0.05	0.05	0.05	0.06	0.12	0.09	0.21	0.05	0.33	0.16	0.17
		P-DORM	0.65	0.06	0.05	0.08	0.09	0.19	0.17	0.28	0.07	0.14	0.11	0.16
		Δ	0.27	0.09	0.07	0.12	0.09	0.13	0.16	0.40	0.05	0.36	0.24	0.24
	DeYO (Lee et al., 2024)	Pure	0.59	0.18	0.06	0.22	0.13	0.23	0.15	4.89	0.01	0.41	0.33	1.16
		P-DORM	0.54	0.19	0.06	0.08	0.20	0.21	0.16	0.21	0.05	0.30	0.12	0.17
		Δ	0.06	0.21	0.05	0.26	0.12	0.14	0.07	4.82	0.05	0.26	0.36	1.11
	CoME (Zhang et al., 2025)	Pure	0.05	0.13	0.01	0.09	0.09	0.08	0.40	0.49	0.03	0.38	0.64	0.39
		P-DORM	0.05	0.17	0.01	0.17	0.08	0.10	0.13	0.51	0.05	0.72	0.41	0.36
		Δ	0.02	0.13	0.02	0.14	0.06	0.07	0.44	0.54	0.06	0.82	0.55	0.48
	BufferTTA (Kim et al., 2025)	Pure	0.09	0.17	0.01	0.13	0.08	0.09	1.01	10.54	0.06	0.30	2.39	2.86
		P-DORM	0.07	0.11	0.04	0.24	0.14	0.12	0.99	0.17	0.06	0.76	0.30	0.46
		Δ	0.05	0.18	0.03	0.14	0.21	0.12	0.91	10.61	0.07	0.64	2.68	2.98
	EATA-C (Tan et al., 2025)	Pure	0.14	0.15	0.03	0.17	0.16	0.13	0.24	0.58	0.36	0.23	0.47	0.38
		P-DORM	0.12	0.10	0.11	0.16	0.21	0.14	0.11	0.26	0.18	0.68	0.30	0.31
		Δ	0.02	0.16	0.14	0.12	0.23	0.14	0.16	0.58	0.31	0.74	0.55	0.46
	RDumb (Press et al., 2023)	Pure	0.35	0.28	0.02	0.10	0.13	0.17	0.34	1.06	0.03	0.30	0.09	0.36
		P-DORM	0.28	0.18	0.04	0.15	0.15	0.16	0.31	2.18	0.04	0.26	0.30	0.62
		Δ	0.09	0.24	0.03	0.09	0.13	0.12	0.12	2.30	0.04	0.18	0.24	0.58
	Across methods	Avg. $\mathrm{Std}(\Delta)$	0.09	0.17	0.06	0.15	0.14	0.12	0.31	3.21	0.10	0.50	0.77	0.98

H.2 Directional and Scale Components of Optimizer State

We use *optimizer history* as an umbrella term for distinct state components. Momentum and first-moment buffers accumulate signed gradients and provide the historical direction used by P-DORM's alignment signal. Second-moment buffers in adaptive optimizers store unsigned gradient-magnitude information and primarily provide coordinate-wise scale preconditioning (Kingma & Ba, 2015; PyTorch Contributors, 2026; Zhuang et al., 2020). Although these scale states do not themselves encode a signed historical direction, stale estimates can miscalibrate post-shift step sizes and thereby alter the resulting update. Our detection analysis concerns incompatibility in the directional state, while the optimizer-specific reset rules may also clear or attenuate associated scale states after a trigger, as detailed in Appendix D.

H.3 Test-Domain Set Notation

In Section 3.1, X^t is a batch-valued observation and $\mathcal{D}^t$ denotes the collection of admissible test batches associated with the current domain. The relation $X^t \in \mathcal{D}^t$ therefore expresses set membership and does not assert that a batch is sampled from a probability distribution denoted by $\mathcal{D}^t$. A probabilistic formulation could instead introduce a separate distribution P^t and write individual samples as $x \sim P^t$, but that distributional notation is not required by the online TTA formulation used in this paper.

I Limitations and Future Work

1. **Small-batch noise.** Extremely small test batches produce unreliable gradients, which can weaken the alignment signal and destabilize reset decisions. Aggregating evidence across batches or using more noise-robust directional estimates may improve detection in this regime.

Table 10: **Seed-wise variability on DomainNet-126.** Definitions and notation follow Table 8.

	Method	Variant	Adam					SGD				
			TTA	CTTA	PTTA	Mix	Avg.	TTA	CTTA	PTTA	Mix	Avg.
(a) ResNet-50	TENT (Wang et al., 2021)	Pure	0.04	0.16	0.18	0.12	0.12	0.05	0.27	0.22	0.15	0.17
		P-DORM	0.07	0.16	0.18	0.14	0.14	0.10	0.22	0.20	0.14	0.16
		Δ	0.03	0.06	0.02	0.09	0.05	0.06	0.17	0.14	0.15	0.13
	DeYO (Lee et al., 2024)	Pure	0.19	0.23	0.39	0.21	0.25	0.20	0.38	0.35	0.16	0.27
		P-DORM	0.15	0.24	0.26	0.20	0.21	0.20	0.36	0.27	0.31	0.29
		Δ	0.08	0.04	0.18	0.05	0.09	0.14	0.16	0.15	0.25	0.18
	CoME (Zhang et al., 2025)	Pure	0.19	0.18	0.18	0.33	0.22	0.11	0.15	0.15	0.15	0.14
		P-DORM	0.23	0.24	0.21	0.37	0.26	0.11	0.13	0.13	0.09	0.11
		Δ	0.08	0.06	0.05	0.13	0.08	0.09	0.03	0.05	0.07	0.06
	BufferTTA (Kim et al., 2025)	Pure	0.15	0.10	0.15	0.15	0.14	0.11	0.11	0.18	0.07	0.12
		P-DORM	0.14	0.11	0.15	0.14	0.13	0.11	0.11	0.17	0.07	0.12
		Δ	0.02	0.03	0.04	0.06	0.03	0.01	0.01	0.01	0.01	0.01
	EATA-C (Tan et al., 2025)	Pure	0.12	0.08	0.23	0.11	0.13	0.12	0.26	0.43	0.30	0.28
		P-DORM	0.11	0.16	0.20	0.14	0.15	0.18	0.27	0.50	0.14	0.27
		Δ	0.05	0.13	0.13	0.05	0.09	0.11	0.20	0.23	0.18	0.18
	RDumb (Press et al., 2023)	Pure	0.18	0.14	0.13	0.16	0.15	0.11	0.21	0.26	0.24	0.20
		P-DORM	0.18	0.14	0.21	0.19	0.18	0.13	0.26	0.27	0.25	0.23
		Δ	0.02	0.05	0.13	0.07	0.07	0.06	0.13	0.14	0.09	0.11
	Across methods	Avg. $\mathrm{Std}(\Delta)$	0.05	0.06	0.09	0.08	0.07	0.08	0.12	0.12	0.13	0.11
(b) ViT-B/16	TENT (Wang et al., 2021)	Pure	0.02	0.00	0.01	0.03	0.01	0.03	0.05	0.05	0.02	0.04
		P-DORM	0.01	0.01	0.03	0.02	0.02	0.05	0.05	0.03	0.01	0.04
		Δ	0.01	0.01	0.02	0.02	0.01	0.06	0.04	0.05	0.03	0.05
	DeYO (Lee et al., 2024)	Pure	0.27	0.12	0.11	0.01	0.13	0.06	0.02	0.01	0.02	0.02
		P-DORM	0.26	0.07	0.15	0.01	0.12	0.03	0.03	0.00	0.02	0.02
		Δ	0.06	0.16	0.05	0.01	0.07	0.04	0.02	0.00	0.02	0.02
	CoME (Zhang et al., 2025)	Pure	0.06	0.10	0.05	0.05	0.06	0.30	0.12	0.11	0.07	0.15
		P-DORM	0.07	0.18	0.07	0.09	0.10	0.28	0.12	0.08	0.10	0.14
		Δ	0.02	0.11	0.10	0.06	0.07	0.13	0.14	0.12	0.06	0.11
	BufferTTA (Kim et al., 2025)	Pure	0.01	0.02	0.01	0.02	0.02	0.01	0.38	0.09	0.37	0.21
		P-DORM	0.01	0.02	0.02	0.03	0.02	0.01	0.05	0.03	0.03	0.03
		Δ	0.02	0.01	0.01	0.01	0.01	0.00	0.39	0.08	0.37	0.21
	EATA-C (Tan et al., 2025)	Pure	0.11	0.04	0.16	0.05	0.09	0.02	0.02	0.05	0.04	0.03
		P-DORM	0.07	0.06	0.17	0.06	0.09	0.02	0.07	0.05	0.05	0.05
		Δ	0.08	0.05	0.10	0.03	0.07	0.02	0.08	0.06	0.08	0.06
	RDumb (Press et al., 2023)	Pure	0.04	0.08	0.14	0.12	0.10	0.08	0.50	0.18	0.17	0.23
		P-DORM	0.03	0.09	0.13	0.13	0.10	0.18	0.38	0.33	0.28	0.29
		Δ	0.03	0.09	0.04	0.08	0.06	0.11	0.54	0.41	0.21	0.32
	Across methods	Avg. $\mathrm{Std}(\Delta)$	0.04	0.07	0.05	0.03	0.05	0.06	0.20	0.12	0.13	0.13

Table 11: **Computational efficiency of P-DORM.** Results are averaged over five optimizers. Params Δ denotes additional trainable parameters, GPU state Δ counts the added detector tensors, and CPU FIFO Δ reports the compact float32 payload of $\mathcal{W}^t$ for $W = 100$. Peak VRAM is measured at MB resolution. Update Δ is the additional optimizer-update time, and step Δ normalizes this increment by the end-to-end Pure step time.

Backbone (L)	Params Δ	GPU state Δ	CPU FIFO Δ	VRAM Δ	Update Δ (ms)	Step Δ
ResNet-50 (166)	0	1.3 KB	66 KB	0 MB	37.32	39.1%
ViT-B/16 (152)	0	1.2 KB	61 KB	0 MB	36.64	18.4%
WideResNet-28-10 (80)	0	0.6 KB	32 KB	0 MB	18.35	11.7%

2. **Conservative thresholds.** The default threshold prioritizes stability and may miss weak or short-lived shifts. Adapting reset sensitivity to the observed stream dynamics could recover additional performance.
3. **Runtime overhead.** Although P-DORM adds negligible memory, the current tensor-wise implementation introduces model-dependent latency. Batched monitoring and kernel fusion may reduce this cost.