Subspace Optimization for Backpropagation-Free Continual Test-Time Adaptation

Anonymous Authors[1]

Abstract

We introduce PACE, a backpropagation-free continual test-time adaptation system that directly optimizes the affine parameters of normalization layers. Existing derivative-free approaches struggle to balance runtime efficiency with learning capacity, as they either restrict updates to input prompts or require continuous, resource-intensive adaptation regardless of domain stability. To address these limitations, PACE leverages the Covariance Matrix Adaptation Evolution Strategy with the Fastfood projection to optimize high-dimensional affine parameters within a low-dimensional subspace, leading to superior adaptive performance. Furthermore, we enhance the runtime efficiency by incorporating an adaptation stopping criterion and a domain-specialized vector bank to eliminate redundant computation. Our framework achieves state-of-the-art accuracy across multiple benchmarks under continual distribution shifts, reducing runtime by over 50% compared to existing backpropagation-free methods. The code is available at https://anonymous.4open.science/r/PACE_.

1. Introduction

Test-time adaptation (TTA) (Wang et al., 2021) has emerged as a practical approach to adapt a deployed neural networks on-the-fly, increasing its robustness to shifting data distributions. While backpropagation (BP)-based methods (Niu et al., 2023; 2022; Wang et al., 2022; 2021) achieve strong performance via self-supervised learning, their high memory requirements and incompatibility with non-differentiable quantized models limit their deployment on resource-constrained edge devices (Niu et al., 2024; Deng et al., 2025; Jia et al., 2024). Conversely, although existing BP-free methods reduce memory overhead and support quantization (Boudiaf et al., 2022; Iwasawa & Matsuo, 2021; Deng et al., 2025; Niu et al., 2024), the inherent challenges of derivative-free adaptation hinder their ability to balance runtime efficiency with learning capacity.

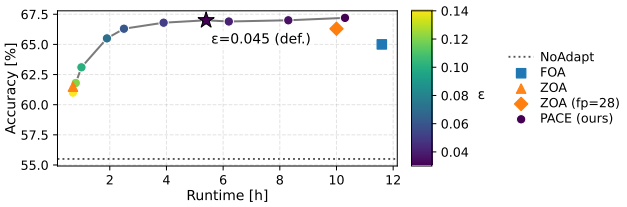

Figure 1. Accuracy versus runtime trade-off on the ImageNet-C benchmark using a ViT-B model, across various adaptation stopping thresholds ϵ (star marks the default setting $\epsilon = 0.045$). The horizontal dotted line represents the NoAdapt baseline accuracy. In addition to outperforming existing baselines and preventing the waste of computational resources on marginal accuracy gains, our approach provides a convenient mechanism to control the accuracy-runtime trade-off.

Existing BP-free methods suffer from two primary limitations: 1) restricted learning capacity: current state-of-the-art does not update the model parameters (Boudiaf et al., 2022; Iwasawa & Matsuo, 2021; Schneider et al., 2020; Khurana et al., 2021; Lim et al., 2023), limits updates to input prompts (Niu et al., 2024), or relies on inherently noisy zeroth-order gradient estimation (Spall, 2002; Deng et al., 2025), which limits their ability to resolve complex distribution shifts. 2) high computational overhead: existing approaches (Niu et al., 2024; Deng et al., 2025) require numerous forward passes to match the performance of BP-based methods. Furthermore, these methods waste resources by performing adaptation on every batch indefinitely, regardless of domain stability.

To address these challenges, we introduce Projected Adaptation via Covariance Evolution (PACE), an efficient BP-free adaptation system that expands learning capabilities while minimizing inference overhead. We start by adapting the model utilizing the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). While CMA-ES is effective for TTA, its high-dimensional complexity typically limits opti-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the ICML 2026 Workshop "Continual Adaptation at Scale: Towards Sustainable AI". Do not distribute.

mization to input prompts (Niu et al., 2024). However, we find that updating the affine parameters of normalization layers, as commonly done in TTA (Wang et al., 2021; Niu et al., 2023; 2022; Döbler et al., 2023), yields significantly better performance (Fig. 2). To make this tractable, PACE leverages the low intrinsic dimensionality of TTA gradients (as noted in (Duan et al., 2025) and Fig. 3) by optimizing a compact vector projected via the Fastfood transform (Le et al., 2013) for efficiency. To further reduce overhead, we leverage the observation that the CMA-ES adaptation predominantly occurs at the onset of a stable domain, yielding negligible accuracy improvements in later batches (Fig. 4) and introduce a dynamic stopping criterion. Finally, we maintain a bank of domain-specialized vectors to enable rapid recovery in recurring similar domains. Together, these components form a practical, backpropagation-free system for long-term edge deployment.

In summary, our contributions are:

- We introduce PACE, a BP-free continual TTA method that enables tractable normalization layer adaptation with CMA-ES by optimizing within a low-dimensional subspace via Fastfood transform.
- We design a dynamic stopping criterion based on the CMA-ES distribution mean to eliminate redundant computation by halting adaptation during stable domain phases.
- We develop a domain-specialized vector bank to facilitate knowledge accumulation and ensure rapid performance recovery in recurring domains.

PACE outperforms existing BP-free approaches in both accuracy and efficiency, reducing runtime by 50% and consistently surpassing the CMA-ES-based FOA baseline.

2. Problem Statement

Continual TTA consists of adapting a pre-trained source model $f_{\boldsymbol{\theta}^{(0)}}$ to a non-stationary sequence of unlabeled domains. The model is initially trained on a labeled source dataset $\mathcal{D}_s = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_s}$ sampled from distribution $\mathcal{P}_s$. At inference, the model encounters a sequence of T domains: $\mathcal{D}_t^{(1)}, \ldots, \mathcal{D}_t^{(T)}$, where $\mathcal{D}_t^{(k)} = \{\boldsymbol{x}_j^{(k)}\}_{j=1}^{N_k}$ and $\boldsymbol{x}_j^{(k)} \sim \mathcal{Q}^{(k)}(\boldsymbol{x})$.

Each domain $\mathcal{D}_t^{(k)}$ consists of unlabeled samples $\{\boldsymbol{x}_j^{(k)}\}$. These domains are out-of-distribution ($\mathcal{Q}^{(k)} \neq \mathcal{P}_s$) and the sequence is non-stationary ($\mathcal{Q}^{(k)} \neq \mathcal{Q}^{(k+1)}$). The model is adapted on-the-fly using only current samples.

3. Methodology

We introduce PACE, which comprises three main components: Subspace Adaptation 3.1, Adaptation Stopping 3.2, and Domain-Specialized Vector Bank 3.3. A more detailed description of the proposed method is provided in the Appendix (Sec. A.3).

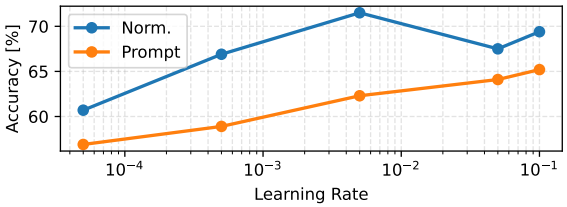

Figure 2. A comparison of updating the affine parameters of normalization layers (**Norm.**) versus three input prompts (**Prompt**) in ViT-B model during test-time adaptation on ImageNet-C benchmark. We evaluate these using ground-truth labels with an SGD optimizer with varying learning rate. Updating the normalization layers allows the model to 'undo' the covariate shift at every depth of the network more effectively for every tested learning rate value.

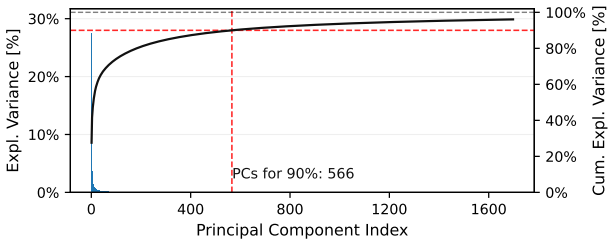

Figure 3. Intrinsic dimensionality of continual TTA gradients in ViT-B. An analysis of concatenated gradients from all ImageNet-C domains reveals that only 566 components explain 90 % of the variance. These results are obtained via optimization of the affine parameters of normalization layers using SGD with the loss function from FOA (Niu et al., 2024). They highlight the low-dimensional nature of the adaptation space. Note that only the most significant components are shown for clarity. The analysis is based on 11 729 gradient samples (batches).

3.1. Subspace Adaptation

Model Update. We adapt the model by adding the constant random projection $\mathrm{proj}(\cdot)$ of a low-dimensional vector $\boldsymbol{v} \in \mathbb{R}^d$ into high-dimensional space $\mathrm{proj}(\boldsymbol{v}) \in \mathbb{R}^D$ to adapted model weights, where $D \gg d$. The adapted model weights are the affine parameters of normalization layers, therefore we set D to be equal to their total dimensionality. We partition $\mathrm{proj}(\boldsymbol{v})$ to match the dimensionality of each parameter tensor and add the resulting components to the initial weights to get the adapted model: $f_{\boldsymbol{\theta}^{(0)}+\mathrm{proj}(\boldsymbol{v})}$. Initializing with a zero vector ($\boldsymbol{v} = \mathbf{0}$) ensures the adaptation starts exactly from the state of the pre-trained model. Our adaptation objective is to find the optimal low-dimensional vector $\boldsymbol{v}^*$, that minimizes the fitness function $\mathcal{L}(\cdot)$, given test samples $\boldsymbol{x}$.

$$\boldsymbol{v}^* = \underset{\boldsymbol{v}}{\operatorname{argmin}}\, \mathcal{L}(f_{\boldsymbol{\theta}^{(0)}+\mathrm{proj}(\boldsymbol{v})}(\boldsymbol{x})) \tag{1}$$

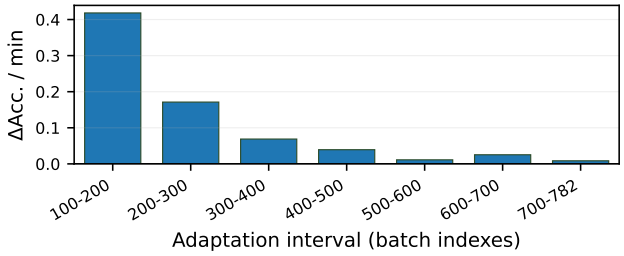

Figure 4. Marginal accuracy gain per unit time across adaptation budgets on ImageNet-C with ViT-B model. For each consecutive pair of adaptation step budgets, we compute the increase in mean accuracy across domains and divide it by the additional estimated wall-clock time required for that interval using our Subspace Adaptation with CMA-ES algorithm. The mean of CMA-ES distribution $\boldsymbol{m}$ is used as the evaluated model update. Higher bars indicate more efficient use of adaptation time, while decreasing bars indicate diminishing returns from further adaptation.

We optimize the $\boldsymbol{v}$ using CMA-ES, which is an evolutionary optimization algorithm that iteratively adapts a multivariate Gaussian distribution by sampling candidate solutions and updating its mean $\boldsymbol{m}^{(t)}$, step size $\tau^{(t)}$, and covariance matrix $\boldsymbol{\Sigma}^{(t)}$ based on the fitness rankings of those samples to progressively increase the likelihood of reaching successful outcomes. We utilize the fitness function used in FOA (Niu et al., 2024) and ZOA (Deng et al., 2025) because of its proven effectiveness and to enable direct comparison. It simultaneously minimizes prediction entropy to increase model confidence and penalize the L_2 distance between test-batch and source-domain activation statistics $(\mu_{s,i}, \sigma_{s,i})$ to maintain feature consistency across L intermediate blocks.

Efficient Dimensionality Expansion. To efficiently project low-dimensional vectors $\boldsymbol{v}$ into the high-dimensional parameter space, we employ the Fastfood transform (Le et al., 2013). It approximates high-dimensional Gaussian random projections by replacing a dense, memory-intensive matrix with a product of structured diagonal, permutation, and Walsh-Hadamard matrices, reducing computational complexity from $O(D^2)$ to $O(D \log D)$. By default, we set $d = 2304$ and the dimensionality of the updated ViT-B model parameters is equal $D = 34800$. In that case, the Fastfood transform significantly reduces memory overhead, requiring only 0.13 MB compared to the 306 MB needed for a standard dense projection matrix.

3.2. Adaptation Stopping

Stopping Heuristic. Because CMA-ES continuously updates its distribution to track optimal candidates, we monitor the relative change in the distribution's mean $\boldsymbol{m}$. Specifically, we stop the adaptation when the normalized difference between the distribution mean at the current time step $\boldsymbol{m}^{(t)}$ and the previous time step $\boldsymbol{m}^{(t-1)}$ falls below a convergence threshold ϵ.

Resuming the Adaptation. Since continual TTA lacks explicit domain labels, we require a criterion to re-initialize the adaptation process. To detect these shifts, we adopt the domain shift detection scheme established in (Hong et al., 2023; Chen et al., 2024; Deng et al., 2025).

3.3. Domain-Specialized Vector Bank

We maintain a memory bank $\mathcal{B}$ of vectors $\boldsymbol{v}$ derived from past domains. Specifically, upon detecting a domain shift, we archive the current mean of the CMA-ES search distribution, $\boldsymbol{m}$, into the bank. For the incoming domain, the new CMA-ES optimizer is initialized using the archived mean that minimizes the fitness function $\mathcal{L}$ on the current data batch $\boldsymbol{x}^{(t)}$:

$$\boldsymbol{m}_{init} = \arg\min_{\boldsymbol{m}_i \in \mathcal{B}} \mathcal{L}(f_{\boldsymbol{\theta}^{(0)} + \text{proj}(\boldsymbol{m}_i)}(\boldsymbol{x}^{(t)})) \tag{2}$$

Adaptation then proceeds conventionally. This retrieval mechanism facilitates the rapid reuse of learned experiences, effectively preventing performance degradation even in the presence of sudden domain shifts.

4. Experiments

4.1. Experimental Details

Datasets and models. We conduct experiments on three standard TTA benchmarks: ImageNet-C (Hendrycks & Dietterich, 2019), ImageNet-R (Hendrycks et al., 2021), and DomainNet-126 (Peng et al., 2019). Experiments are conducted with both full-precision and quantized versions of ViT-B (Dosovitskiy et al., 2021) and DeiT-B (Touvron et al., 2021) models. Unless otherwise specified, experiments are reported with full-precision models.

Baselines. We compare against both previously defined categories of state-of-the-art TTA methods: BP-free, including LAME (Boudiaf et al., 2022), T3A (Iwasawa & Matsuo, 2021), FOA (Niu et al., 2024), and ZOA (Deng et al., 2025), and BP-based, specifically TENT (Wang et al., 2021), CoTTA (Wang et al., 2022), and SAR (Niu et al., 2023). To ensure a fair comparison, we also compare with ZOA (fp=28), a modified version of ZOA where we match the number of forward passes used in FOA and PACE by increasing the sampled perturbations for gradient estimation to 27. Further details and hyperparameter ablation studies are provided in Appendix Secs. A.4 and A.5.4, respectively

4.2. Results

Results on Full-Precision Models. PACE achieves the highest average accuracy across all benchmarks and full-precision models, consistently outperforming other BP-free baselines (Tab. 1). Specifically, it surpasses direct competitors FOA and ZOA by 1.7 and 4.0 percentage points, respectively. When ZOA is modified to match the forward-

Table 1. Accuracy on ImageNet-C (IN-C), ImageNet-R (IN-R) and DomainNet-126 (DN) with ViT-B and DeiT-B. **Bold** indicates best result, underlined second best.

		ViT-B			DeiT-B			
Method	Backprop.	IN-C	IN-R	DN	IN-C	IN-R	DN	**Avg.**
NoAdapt	✗	55.5	59.5	53.1	51.6	52.8	60.2	55.5
TENT	✔	61.7	63.9	53.8	55.5	56.1	60.4	58.6
CoTTA	✔	58.4	63.5	**62.0**	55.4	53.0	60.4	58.8
SAR	✔	61.5	63.3	53.8	59.4	57.4	60.7	59.4
LAME	✗	54.1	59.0	51.6	50.9	52.5	58.9	54.5
T3A	✗	55.4	58.0	56.2	43.5	49.7	61.8	54.1
FOA	✗	65.0	63.8	56.0	61.3	56.3	63.0	60.9
ZOA	✗	61.5	60.7	55.8	56.9	53.5	62.4	58.5
ZOA (fp=28)	✗	66.3	62.6	56.5	61.7	55.8	63.4	61.1
PACE (ours)	✗	**67.0**	**64.5**	57.0	**62.7**	**59.5**	**64.3**	**62.5**

Table 2. Corruption Accuracy (%) on ImageNet-C with **Quantized** ViT-B models.

Model	NoAdapt	T3A	FOA	ZOA	ZOA (fp=28)	PACE (ours)
8-bit	54.1	55.6	63.5	59.7	64.7	**65.0**
6-bit	47.7	43.3	55.8	54.3	57.5	**58.1**

pass budget of PACE and FOA, our method maintains a 1.4 percentage point lead despite adapting to fewer batches (Tab. A.1). While CoTTA achieves the highest accuracy on DomainNet-126 by leveraging backpropagation, it is significantly more memory-intensive than PACE (Tab. A.1) and lacks support for quantized model updates. Despite the constraints of BP-free optimization, PACE outperforms BP-based methods on average, underscoring the effectiveness of our approach. More detailed results for ImageNet-C are presented in Tab. A.2 and Fig. A.2 in the Appendix.

Results on Quantized Models Our method consistently outperforms competing approaches on quantized models across both bit-widths (Tab. 2). Notably, our 8-bit model achieves 65.0 % accuracy, matching the 32-bit FOA baseline while significantly exceeding almost all 32-bit competitors. The performance margin over ZOA (fp = 28) increases as quantization becomes more aggressive, rising from 0.3 percentage points at 8-bit to 0.6 percentage points at 6-bit.

Ablation Study Tab. 3 decomposes the performance gains of PACE across its three core components. To isolate the impact of subspace adaptation, we first compare PACE v1 against FOA. While both methods employ CMA-ES, FOA updates input prompts whereas PACE v1 exclusively utilizes subspace adaptation. PACE v1 outperforms FOA by 1.1 percentage points, demonstrating that the low-dimensional update of normalization layers is more effective than input prompt tuning. Integrating the domain-specialized vector bank and shift detection (PACE v2) further improves accuracy. This suggests that the model effectively reuses fine-tuned vectors from previously encountered similar domains. This configuration achieves the highest overall result by maintaining updates for every data point. In contrast, PACE v3 prioritizes efficiency by utilizing adaptation stopping but not the vector bank, which reduces runtime.

Table 3. Ablation study on each component of PACE on ImageNet-C with ViT-B.

Method	adapt. stopping	vector bank	subspace adapt.	Avg. Acc.	Runtime (hours)
NoAdapt				55.5	0.01
Baseline (FOA)				65.0	11.6
PACE v1			✔	66.1	10.3
PACE v2		✔	✔	67.2	10.3
PACE v3	✔		✔	66.3	5.4
PACE	✔	✔	✔	67.0	5.4

Ultimately, the full PACE provides the optimal trade-off, combining all three components to achieve high accuracy with low computational overhead.

Computational Complexity Analysis. Fig. 1 shows the accuracy-runtime trade-off for varying adaptation threshold ϵ. Additionally, Tab. A.1 in the Appendix compares the wall-clock time, memory consumption, and percentage of adaptation batches of PACE against BP-based and BP-free baselines. While PACE maintains the low memory footprint typical of BP-free methods, it improves efficiency over leading alternatives such as FOA and ZOA (fp = 28). Specifically, our adaptation stopping technique reduces their wall-clock time by 53 % and 46 %, respectively, cutting runtime from over 10 h to 5.4 h while simultaneously increasing accuracy. When we increase the threshold ϵ to match the runtime of default ZOA (PACE with $\epsilon \in \{0.125, 0.14\}$), accuracy gains diminish to levels similar to ZOA or lower. However, PACE achieves this performance by adapting to only 3.5 %–4.8 % of batches, requiring only a single forward pass for the remaining samples. For PACE (K=6), we reduce the population size and maintain the default ϵ, which outperforms ZOA by 0.3 percentage points and reduces runtime by an additional 0.1 h, despite adapting to 13.7 % of batches. These results indicate that the adaptation stopping has its limits and keeping the ϵ threshold low while decreasing population size is a more effective strategy for drastically minimizing total runtime. Figs. 1 and A.3 in the Appendix further validate the efficiency of our approach on the JETSON XAVIER NX.

5. Conclusions

In this work, we presented PACE, an efficient BP-free framework for continual test-time adaptation. By exploiting the low intrinsic dimensionality of normalization layer updates in TTA, PACE overcomes the restricted learning capacity of prior BP-free methods without the prohibitive memory requirements of backpropagation. Furthermore, the introduction of an adaptation stopping criterion and a domain-specialized vector bank ensures that PACE remains efficient and robust during long-term deployment across recurring distribution shifts. We experimentally verified that our approach enables high-performance adaptation on resource-constrained edge devices.

References

Boudiaf, M., Mueller, R., Ben Ayed, I., and Bertinetto, L. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8344–8353, 2022.

Chen, G., Niu, S., Chen, D., Zhang, S., Li, C., Li, Y., and Tan, M. Cross-device collaborative test-time adaptation. *Advances in Neural Information Processing Systems*, 37: 122917–122951, 2024.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Deng, Z., Chen, G., Niu, S., Luo, H., Zhang, S., Yang, Y., Chen, R., Luo, W., and Tan, M. Test-time model adaptation for quantized neural networks. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 7258–7267, 2025.

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

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Duan, D., Xu, R., Liu, P., and Wen, F. Lifelong test-time adaptation via online learning in tracked low-dimensional subspace. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Gong, T., Jeong, J., Kim, T., Kim, Y., Shin, J., and Lee, S.-J. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

Hansen, N. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021.

Hoang, T. H., Vo, M., and Do, M. Persistent test-time adaptation in recurring testing scenarios. *Advances in Neural Information Processing Systems*, 37:123402–123442, 2024.

Hong, J., Lyu, L., Zhou, J., and Spranger, M. Mecta: Memory-economic continual test-time model adaptation. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Jia, H., Kwon, Y., Orsino, A., Dang, T., Talia, D., and Mascolo, C. Tinytta: Efficient test-time adaptation via early-exit ensembles on edge devices. *Advances in Neural Information Processing Systems*, 37:43274–43299, 2024.

Khurana, A., Paul, S., Rai, P., Biswas, S., and Aggarwal, G. Sita: Single image test-time adaptation. *arXiv preprint arXiv:2112.02355*, 2021.

Le, Q., Sarlós, T., Smola, A., et al. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85, 2013.

Lim, H., Kim, B., Choo, J., and Choi, S. Ttn: A domain-shift aware batch normalization in test-time adaptation. *arXiv preprint arXiv:2302.05155*, 2023.

Mirza, M. J., Masana, M., Possegger, H., and Bischof, H. An efficient domain-incremental learning approach to drive in all weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3001–3011, 2022.

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

Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., and Tan, M. Efficient test-time model adaptation without forgetting. In *ICML*, volume 162, pp. 16888–16905, 2022.

Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., and Tan, M. Towards stable test-time adaptation in dynamic wild world. In *Internetional Conference on Learning Representations*, 2023.

Niu, S., Miao, C., Chen, G., Wu, P., and Zhao, P. Test-time model adaptation with only forward passes. In *The International Conference on Machine Learning*, 2024.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. Improving robustness against common corruptions by covariate shift adaptation. *Advances in neural information processing systems*, 33: 11539–11551, 2020.

Spall, J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 2002.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Tan, M., Chen, G., Wu, J., Zhang, Y., Chen, Y., Zhao, P., and Niu, S. Uncertainty-calibrated test-time model adaptation without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jegou, H. Training data-efficient image transformers & distillation through attention. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

Vray, G., Tomar, D., Gao, X., Thiran, J.-P., Shelhamer, E., and Bozorgtabar, B. Reservoirtta: Prolonged test-time adaptation for evolving and recurring domains. *arXiv preprint arXiv:2505.14511*, 2025.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Wang, Q., Fink, O., Van Gool, L., and Dai, D. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7201–7211, 2022.

Wightman, R. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Yuan, Z., Xue, C., Chen, Y., Wu, Q., and Sun, G. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *European conference on computer vision*, pp. 191–207. Springer, 2022.

Zhang, Y., Gong, K., Ding, X., Zhang, K., Lv, F., Keutzer, K., and Yue, X. Towards unified and effective domain generalization. 2023.

Zhao, B., Chen, C., and Xia, S.-T. Delta: degradation-free fully test-time adaptation. *arXiv preprint arXiv:2301.13018*, 2023.

Appendix

A.1. Overview

This Appendix provides related work section (Sec. A.2), detailed description of the proposed method (Sec A.3), comprehensive experimental details (Sec. A.4) and additional experiments (Sec. A.5). Experiments include the computational complexity analysis (Sec. A.5.1), detailed results on the ImageNet-C benchmark (Sec. A.5.2), recurring domains scenario (Sec. A.5.3) and hyperparameter ablation studies (Sec A.5.4) on the optimization dimensionality d, capacity of domain-specialized vector bank p, and domain shift threshold γ.

A.2. Related Work

Test-Time Adaptation. TTA targets distribution shifts during inference by adapting pre-trained models without access to source data or labels (Wang et al., 2021). Continual TTA further assumes a non-stationary environment where domains change without access to domain labels (Wang et al., 2022). Existing methods can be divided into backpropagation-based (BP-based) and backpropagation-free (BP-free) approaches.

BP-based methods. These methods utilize gradient descent to update model parameters at test time. To balance efficiency with learning capacity, most approaches restrict updates to affine parameters in normalization layers (Niu et al., 2023; 2022; Wang et al., 2021; Döbler et al., 2023). In the absence of labels, these methods employ self-supervised objectives such as entropy minimization (Wang et al., 2021; Niu et al., 2022; Tan et al., 2025), pseudo-labeling variants (Döbler et al., 2023; Wang et al., 2022), rotation prediction (Sun et al., 2020), or feature distribution alignment (Mirza et al., 2022). Because self-supervised signals can be unreliable, stabilization techniques are often required. For instance, SAR (Niu et al., 2023) filters samples by entropy and seeks flat loss minima, while CoTTA (Wang et al., 2022) adopts a teacher-student framework with stochastic weight restoration. Despite their effectiveness, BP-based methods demand high memory and differentiable weights, which limits their utility on quantized or resource-constrained edge devices.

BP-free methods. Early BP-free TTA focused on adjusting batch normalization statistics using test data batches (Nado et al., 2020; Schneider et al., 2020; Gong et al., 2022). Subsequent work extended the statistics update to single-sample adaptation (Khurana et al., 2021) and handling the temporal class correlation (Gong et al., 2022; Zhao et al., 2023). However, these approaches require the presence of batch normalization layers in model architectures. Further works explored prototype-based classifier adjustment (T3A) (Iwasawa & Matsuo, 2021) and logits correction (LAME) (Boudiaf et al., 2022). None of those methods update core model weights, resulting in significantly limited learning capacity. Recent work introduces weight-updating BP-free methods: FOA (Niu et al., 2024) adapts input prompts via CMA-ES, and ZOA (Deng et al., 2025) employs zeroth-order gradient estimation. While promising, FOA's prompt-only updates restrict its flexibility. Furthermore, FOA adapts on every batch, demanding up to 28 forward passes, which is computationally prohibitive for every data sample in real-world deployment. In contrast, ZOA reliably updates core model weights with only 2 forward passes. However, to remain competitive with BP-based methods, its computational demands are similar to FOA. Our PACE improves adaptation effectiveness through subspace adaptation of affine parameters of normalization layers via CMA-ES and minimizes overhead by introducing an automated stopping criterion that halts adaptation once it is no longer beneficial. Additionally, drawing on recent BP-based TTA work (Vray et al., 2025), we incorporate a domain-specialized vector bank to aggregate knowledge across diverse environments. This mechanism enables the model to rapidly recover performance when re-encountering domains, significantly enhancing its readiness for practical deployment.

A.3. Methodology

We introduce a novel BP-free continual TTA method, coined Projected Adaptation via Covariance Evolution (PACE), designed to be a fully practical TTA system. It comprises three main components: Subspace Adaptation A.3.1, Adaptation Stopping A.3.2, and Domain-Specialized Vector Bank A.3.3.

A.3.1. Subspace Adaptation

Building on established TTA frameworks (Wang et al., 2021; Niu et al., 2023; 2022; Döbler et al., 2023) and our observation that input prompt tuning is less effective than updating affine parameters of normalization layers (Fig. 2), we employ CMA-ES to optimize these layers. To ensure high-dimensional optimization remains tractable by CMA-ES, we leverage the observation that TTA gradients have low intrinsic dimensionality (as noted in (Duan et al., 2025) and Fig. 3), which suggests

that effective updates can be achieved within a low-dimensional subspace.

Model Update. We adapt the model by adding the constant random projection $\mathrm{proj}(\cdot)$ of a low-dimensional vector $\boldsymbol{v} \in \mathbb{R}^d$ into high-dimensional space $\mathrm{proj}(\boldsymbol{v}) \in \mathbb{R}^D$ to adapted model weights, where $D \gg d$. The adapted model weights are the affine parameters of normalization layers, therefore we set D to be equal to their total dimensionality. We partition $\mathrm{proj}(\boldsymbol{v})$ to match the dimensionality of each parameter tensor and add the resulting components to the initial weights to get the adapted model: $f_{\boldsymbol{\theta}^{(0)}+\mathrm{proj}(\boldsymbol{v})}$. Initializing with a zero vector ($\boldsymbol{v} = \mathbf{0}$) ensures the adaptation starts exactly from the state of the pre-trained model. Our adaptation objective is to find the optimal low-dimensional vector $\boldsymbol{v}^*$, that minimizes the fitness function $\mathcal{L}(\cdot)$, given a test samples $\boldsymbol{x}$.

$$\boldsymbol{v}^* = \underset{\boldsymbol{v}}{\operatorname{argmin}}\, \mathcal{L}(f_{\boldsymbol{\theta}^{(0)}+\mathrm{proj}(\boldsymbol{v})}(\boldsymbol{x})) \tag{3}$$

CMA-ES Optimization. We perform the adaptation using the CMA-ES strategy. Rather than directly optimizing the vector $\boldsymbol{v}$, CMA-ES maintains and optimizes a multivariate Gaussian distribution over the search space. At each iteration t (t-th test batch), a population of K candidate solutions is sampled from ths distribution:

$$\boldsymbol{v}_k^{(t)} \sim \boldsymbol{m}^{(t)} + \tau^{(t)}\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}^{(t)}) \tag{4}$$

where $k = 1, \ldots, K$. Here, $\boldsymbol{m}^{(t)} \in \mathbb{R}^d$ represents the mean of the search distribution at iteration step t, $\tau^{(t)} \in \mathbb{R}^+$ is the overall standard deviation controlling the global step size, and $\boldsymbol{\Sigma}^{(t)}$ is the covariance matrix defining the distribution's shape and orientation.

For each candidate $\boldsymbol{v}_k^{(t)}$, we evaluate its fitness by updating the model and computing the loss on the current test sample $\boldsymbol{x}^{(t)}$. This yields a fitness value l_k for each candidate. CMA-ES then updates the distribution parameters ($\boldsymbol{m}^{(t)}$, $\tau^{(t)}$, and $\boldsymbol{\Sigma}^{(t)}$) for the next generation based on the ranking of the fitness values. This process systematically increases the likelihood of previously successful candidates (see (Hansen, 2016) for details). Following (Niu et al., 2024), we output the prediction associated with the lowest fitness value as the final prediction of the model.

Fitness Function. We utilize the fitness function used in FOA (Niu et al., 2024) and ZOA (Deng et al., 2025) because of its proven effectiveness and to enable direct comparison. Prior to adaptation, we pass a small set of source data through the model to calculate the means and standard deviations of activations from L intermediate model blocks, denoted as $\{\mu_{s,i}, \sigma_{s,i}\}_{i=1}^{L}$. During test time, we calculate the corresponding statistics on the current batch of test samples $\boldsymbol{x}^{(t)}$, yielding $\{\mu_i(\boldsymbol{x}^{(t)}), \sigma_i(\boldsymbol{x}^{(t)})\}_{i=1}^{L}$. The fitness function combines the prediction entropy with the divergence between these activation statistics:

$$\begin{aligned}\mathcal{L}(f_{\boldsymbol{\theta}^{(0)}+\mathrm{proj}(\boldsymbol{v}_k^{(t)})}(\boldsymbol{x}^{(t)})) = \frac{1}{B \times C} \sum_{\boldsymbol{x} \in \boldsymbol{x}_t} \sum_{c=1}^{C} -y_c \log y_c \\ +\lambda \sum_{i=1}^{L} \Big(\|\mu_i(\boldsymbol{x}^{(t)}) - \mu_{s,i}\|_2 + \|\sigma_i(\boldsymbol{x}^{(t)}) - \sigma_{s,i}\|_2 \Big)\end{aligned} \tag{5}$$

where y_c is the c-th element of the prediction probability vector $\hat{\boldsymbol{y}} = f_{\boldsymbol{\theta}^{(0)}+\mathrm{proj}(\boldsymbol{v}_k^{(t)})}(\boldsymbol{x}^{(t)})$, C is the total number of classes, λ is a balancing hyperparameter, and B is the batch size.

Efficient Dimensionality Expansion. To efficiently project low-dimensional vectors $\boldsymbol{v}$ into the high-dimensional parameter space, we employ the Fastfood transform (Le et al., 2013). Standard dimensionality expansion requires multiplying $\boldsymbol{v}$ by a dense projection matrix $\mathbf{W} \in \mathbb{R}^{D \times d}$. Storing this dense matrix can incur high memory costs for large networks.

Fastfood circumvents this bottleneck by approximating the dense Gaussian matrix $\mathbf{W}$ with a product of structured diagonal matrices and the Fast Walsh-Hadamard Transform (FWHT). We redefine the linear projection as:

$$\mathbf{W}\boldsymbol{v} \approx \mathbf{SHGPHB}\boldsymbol{v} = \mathrm{proj}(\boldsymbol{v}) \tag{6}$$

where $\mathbf{B} \in \mathbb{R}^{d \times d}$ is a diagonal matrix with entries sampled uniformly from $\{-1, 1\}$, $\mathbf{P} \in \mathbb{R}^{D \times D}$ is a random permutation matrix, and $\mathbf{G} \in \mathbb{R}^{D \times D}$ is a diagonal matrix with entries drawn from a standard normal distribution $\mathcal{N}(0, 1)$. The matrix $\mathbf{S} \in \mathbb{R}^{D \times D}$ is a diagonal scaling matrix ensuring the rows of the resulting pseudo-random matrix possess the correct L_2 norm to approximate the χ-distribution of a true Gaussian random matrix. Finally, $\mathbf{H}$ represents the Walsh-Hadamard matrix. We perform multiplication by $\mathbf{H}$ via the FWHT, entirely avoiding the instantiation of the matrix in memory.

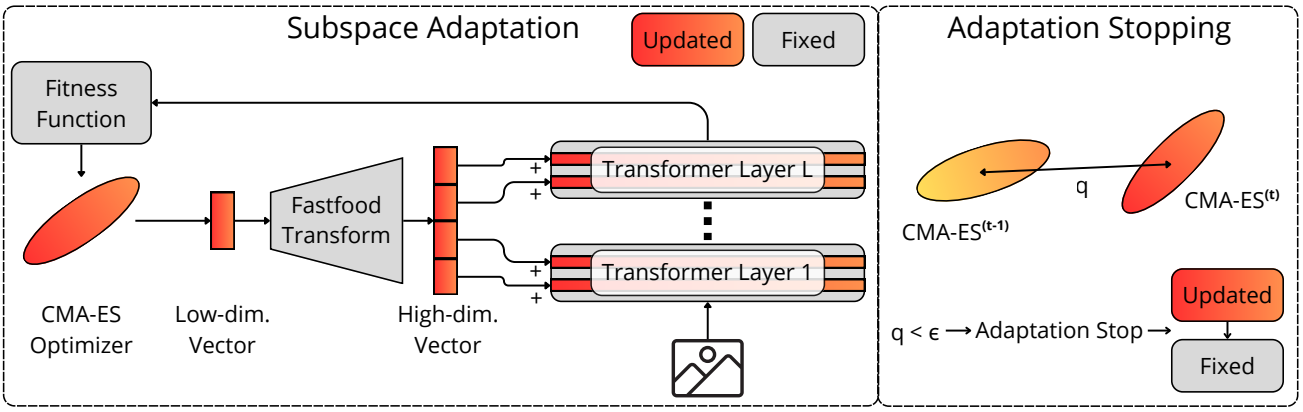

Figure A.1. Diagram of our method. **1) Subspace Adaptation**: We adapt the model by adding a high-dimensional random projection of a small, learnable vector to the model's normalization layer weights. We use the CMA-ES strategy to iteratively evolve a population of these vectors, selecting the one that minimizes the loss on current test samples. **2) Adaptation Stopping**: For efficiency, we stop the adaptation when the mean of the distribution optimized by CMA-ES is lower than the threshold. Along with Domain-Specialized Vector Bank, they make an effective and efficient TTA system.

To achieve the exact target dimensionality D, we pad the input to the nearest power of two (enabling FWHT), process it through stacked independent Fastfood blocks, and subsequently slice it to the required length. We initialize the Fastfood transform components once at the beginning of the adaptation phase and freeze them for the remainder of the process.

By default, we set $d = 2304$ and the dimensionality of the updated ViT-B model parameters is equal $D = 34800$. In that case, the Fastfood transform significantly reduces memory overhead, requiring only 0.13 MB compared to the 306 MB needed for a standard dense projection matrix.

A.3.2. Adaptation Stopping

We observe that adaptation at the beginning of a stable domain drives the majority of performance improvements (Fig. 4). However, the adaptation process incurs significant computational overhead. While test data distributions might remain stable for extended periods in real-world deployment, current BP-free TTA methods adapt indefinitely. This approach wastes computational resources and energy, a critical bottleneck for long-term deployments on embedded devices. To address this inefficiency, we introduce a heuristic to halt adaptation when it yields marginal performance gains.

Stopping Heuristic. Because CMA-ES continuously updates its distribution to track optimal candidates, we monitor the relative change in the distribution's mean $\boldsymbol{m}$. Specifically, we stop the adaptation when the normalized difference between the distribution mean at the current time step $\boldsymbol{m}^{(t)}$ and the previous time step $\boldsymbol{m}^{(t-1)}$ falls below a convergence threshold ϵ:

$$\frac{\|\boldsymbol{m}^{(t)} - \boldsymbol{m}^{(t-1)}\|}{\|\boldsymbol{m}^{(t-1)}\|} < \epsilon \tag{7}$$

We rely on this specific formulation because the mean $\boldsymbol{m}^{(t)}$ represents the algorithm's current best estimate for the optimal parameters $\boldsymbol{v}$. When this relative change approaches zero, it indicates that the CMA-ES optimization has converged on a local optimum for the current data distribution. Furthermore, normalizing the difference by the previous mean ensures that our stopping criterion is scale-invariant, making it robust across different network layers or parameter magnitudes. Once the stopping criterion in Eq. 7 is satisfied, the model is updated with the CMA-ES mean $\boldsymbol{m}^{(t)}$ and fixed.

Resuming the Adaptation. Since continual TTA lacks explicit domain labels, we require a criterion to re-initialize the adaptation process. To detect these shifts, we adopt the domain shift detection scheme established in (Hong et al., 2023; Chen et al., 2024; Deng et al., 2025), which remains invariant to updates to the model.

We maintain the exponential moving average (EMA) of the activation statistics from the stem layer of the model:

$$\phi_{EMA}^{(t-1)} = \beta\phi^{(t-1)} + (1-\beta)\phi_{EMA}^{(t-2)}, \tag{8}$$

where $\phi^{(t-1)}$ represents the activation mean and variance for the $(t-1)$-th batch, and $\beta = 0.8$ denotes the moving average factor. To quantify the shift, we compute the symmetric Kullback-Leibler (KL) divergence between the current batch

statistics $\phi^{(t)}$ and the historical EMA $\phi_{EMA}^{(t-1)}$:

$$U(\phi_{EMA}^{(t-1)}, \phi^{(t)}) = \frac{1}{H}\sum_{i=1}^{H}[KL(\phi_{EMA}^{(t-1),i} \| \phi^{(t),i}) + KL(\phi^{(t),i} \| \phi_{EMA}^{(t-1),i})], \tag{9}$$

where H is the dimensionality of the statistics. A domain shift is detected when this distance exceeds a predefined threshold γ, at which point we re-initialize the adaptation using the Domain-Specialized Vector Bank. To ensure robust detection, we cease updating ϕ_{EMA} once the adaptation has stopped, by the technique described above.

A.3.3. Domain-Specialized Vector Bank

In real-world, long-term deployments, domain recurrence can be a common phenomenon (Vray et al., 2025). A fully practical TTA system must account for this by rapidly reusing knowledge acquired from previously encountered domains.

To achieve this, we maintain a memory bank $\mathcal{B}$ of vectors $\boldsymbol{v}$ derived from past domains. Specifically, upon detecting a domain shift, we archive the current mean of the CMA-ES search distribution, $\boldsymbol{m}$, into the bank. For the incoming domain, the new CMA-ES optimizer is initialized using the archived mean that minimizes the fitness function $\mathcal{L}$ on the current data batch $\boldsymbol{x}^{(t)}$:

$$\boldsymbol{m}_{init} = \arg\min_{\boldsymbol{m}_i \in \mathcal{B}} \mathcal{L}(f_{\boldsymbol{\theta}^{(0)} + \mathrm{proj}(\boldsymbol{m}_i)}(\boldsymbol{x}^{(t)})) \tag{10}$$

Adaptation then proceeds conventionally. This retrieval mechanism facilitates the rapid reuse of learned experiences, effectively preventing performance degradation even in the presence of sudden domain shifts.

To ensure strict memory bounds, we constrain the maximum capacity of the bank to p vectors. When a newly optimized mean is added to a full bank ($|\mathcal{B}| > p$), we employ a redundancy-based removal policy. We calculate the average pairwise cosine similarity for each mean $\boldsymbol{m}_i$ in the bank and discard the vector that exhibits the highest average similarity to the others:

$$\boldsymbol{m}_{drop} = \arg\max_{\boldsymbol{m}_i \in \mathcal{B}} \frac{1}{|\mathcal{B}| - 1} \sum_{\boldsymbol{m}_j \in \mathcal{B} \setminus \{\boldsymbol{m}_i\}} \frac{\boldsymbol{m}_i \cdot \boldsymbol{m}_j}{\|\boldsymbol{m}_i\| \|\boldsymbol{m}_j\|} \tag{11}$$

This strategy effectively prunes the most redundant information, maximizing the diversity of the domain representations stored in the bank.

Subspace adaptation enables memory-efficient storage of knowledge from previous domains. A single float32 domain vector with our default dimensionality ($d = 2304$) occupies approximately $0.0088\,\mathrm{MB}$ of memory. Consequently, the total memory usage is only about $0.26\,\mathrm{MB}$ when the bank's maximum default capacity $p = 30$ is utilized.

A.4. Experimental Details

A.4.1. Datasets and models.

We conduct experiments on three standard TTA benchmarks: ImageNet-C (Hendrycks & Dietterich, 2019), ImageNet-R (Hendrycks et al., 2021), and DomainNet-126 (Peng et al., 2019). ImageNet-C consists of 15 corruption functions across five severity levels. Following the protocol in (Wang et al., 2022), we evaluate using the classic corruption sequence with the highest severity level. ImageNet-R provides diverse renditions of 200 ImageNet classes, while DomainNet-126 contains images from four distinct domains (real, clipart, painting, and sketch).

Experiments are reported with both full-precision and quantized versions of ViT-B (Dosovitskiy et al., 2021) and DeiT-B (Touvron et al., 2021) models. Unless otherwise specified, experiments are reported with full-precision models. We implement quantization using PTQ4ViT (Yuan et al., 2022), following (Niu et al., 2024). For ImageNet benchmarks, we use checkpoints trained on the ImageNet-1K (Deng et al., 2009) training set obtained from the `timm` repository (Wightman, 2019). For DomainNet-126, we utilize models trained on the *real* domain using the repository from (Zhang et al., 2023) and evaluate on the remaining three domains.

A.4.2. Implementation Details

We adopt the hyperparameter settings specified in the original papers for all baselines, except where they were not provided. In those instances, the learning rate was specifically tuned for our model and experimental setup. We utilized method

implementations from the code repository of FOA (Niu et al., 2024) and ZOA (Deng et al., 2025). In the following, we present the details regarding each method.

PACE (ours). To ensure a fair comparison with FOA, we configure the CMA-ES population size K to 28 and set the optimization vector dimensionality d to 2304. Following FOA and ZOA, we utilize the validation set of ImageNet-1K to compute the statistics of in-distribution data, setting λ to 0.2 for ImageNet-R and 0.4 for all other benchmarks. We set the domain shift detection threshold γ to 0.03, the adaptation stopping threshold ϵ to 0.045, and the maximum capacity of the domain-specialized vector buffer p to 30. Our method specifically updates the affine parameters of the normalization layers. Following ZOA, we keep the layer normalization parameters of the first block and the last three blocks of the tested models fixed.

FOA (Niu et al., 2024). We set the number of input prompt embeddings to 3 and the population size K to 28. In-distribution statistics are computed using the ImageNet-1K validation set. The loss trade-off parameter λ is set to 0.2 for ImageNet-R and 0.4 for all other benchmarks, while the moving average factor for batch-to-source shift activation is maintained at 0.1.

ZOA (Deng et al., 2025). The learnable parameters are perturbed with a step size of 0.02 for gradient estimation, while the step size for the coefficients of different domain parameters is set to 0.05. The SGD optimizer with a weight decay of 0.4 is used to update the model parameters, and the AdamW optimizer with a weight decay of 0.1 is used to update the coefficients. The maximum number of domain knowledge parameters is set to 32. The learning rate for coefficients is set to 0.01 for all setups. The learning rate for model parameters is set to 0.0002 for 6-bit ViT-B and 0.0005 for all other models. In terms of ZOA ($fp=28$), the optimal learning rate chosen on ImageNet-C is set to 0.005 for all models and datasets.

LAME (Boudiaf et al., 2022). Following (Niu et al., 2024), we use the kNN affinity matrix set to 5, as this value was found to be optimal for ImageNet-C.

T3A (Iwasawa & Matsuo, 2021). Following (Niu et al., 2024), the number of supports to restore M is set to 20, as this value was found to be optimal for ImageNet-C.

TENT (Wang et al., 2021). We use SGD optimizer, with a momentum of 0.9. The learning rate was tuned on ImageNet-C and set to 0.0001 for both ViT-B and DeiT-B models.

SAR (Niu et al., 2023). We use SGD optimizer with a momentum of 0.9. The learning rate tuned on ImageNet-C is set to 0.001 for both ViT-B and DeiT-B. The entropy threshold E_0 is set to $0.4 \times lnC$, where C is the number of task classes.

CoTTA (Wang et al., 2022). We use SGD optimizer, with a momentum of 0.9. The learning rate was tuned on ImageNet-C and set to 0.001 for ViT-B and 0.005 for DeiT-B. The augmentation threshold p_{th} is set to 0.1. The restoration probability is set to 0.01 and the EMA factor for teacher update is set to 0.999.

A.5. Additional Experimental Results

A.5.1. Computational Complexity

Tab. A.1 compares the wall-clock time, memory consumption, and percentage of adaptation batches of PACE against BP-based and BP-free baselines. While PACE maintains the low memory footprint typical of BP-free methods, it improves efficiency over leading alternatives such as FOA and ZOA (fp$=28$).

A.5.2. Detailed ImageNet-C Results

Fig. A.2 illustrates the per-batch accuracy and throughput on the ImageNet-C dataset using a JETSON XAVIER NX. PACE dynamically leverages throughput by prioritizing adaptation at the onset of new domains (resulting in temporary throughput drops) and resuming high-speed inference once the domain stabilizes. In contrast, competing approaches maintain significantly lower throughput while achieving inferior accuracy (see Tab. A.2). While certain domain shifts remain undetected due to inter-domain similarity, this suggests that re-initiating adaptation in such instances is unnecessary.

A.5.3. Performance On Recurring Domains

Tab. A.3 compares PACE against BP-free approaches in long-term continual adaptation scenarios involving recurring domains, following (Niu et al., 2024; Vray et al., 2025; Hoang et al., 2024). We evaluate performance on the repeated ImageNet-C. While competing methods require multiple passes over the benchmark to converge toward our accuracy,

Table A.1. Computation complexity comparison on ImageNet-C with ViT-B. Forward and backward passes (#FP / #BP) are counted for processing a single sample. The wall-clock time (hours) and memory usage (MB) are measured for processing the whole ImageNet-C on a single RTX 4090 GPU. Adapted batches are indicated by the percentage on which adaptation was performed.

Method	Backprop.	#FP	#BP	Avg. Acc. (%)	Runtime (hours)	Memory (MB)	Adapt. Batches (%)
NoAdapt	✗	1	0	55.5	0.01	819	0
TENT	✔	1	1	61.7	0.03	5,165	100
SAR	✔	[1, 2]	[0, 2]	61.5	1.1	5,166	100
CoTTA	✔	3 or 35	1	58.4	1.5	16,836	100
T3A	✗	1	0	56.9	0.7	957	100
FOA	✗	28	0	65.0	11.6	832	100
ZOA	✗	2	0	61.5	0.7	858	100
ZOA (fp = 28)	✗	28	0	66.3	10.0	862	100
PACE ($\epsilon = 0.125$)	✗	1 or 28	0	61.5	0.8	863	4.8
PACE ($\epsilon = 0.14$)	✗	1 or 28	0	61.0	0.7	863	3.5
PACE ($K = 6$)	✗	1 or 6	0	61.8	0.6	863	13.7
PACE	✗	1 or 28	0	**67.0**	5.4	863	50.6

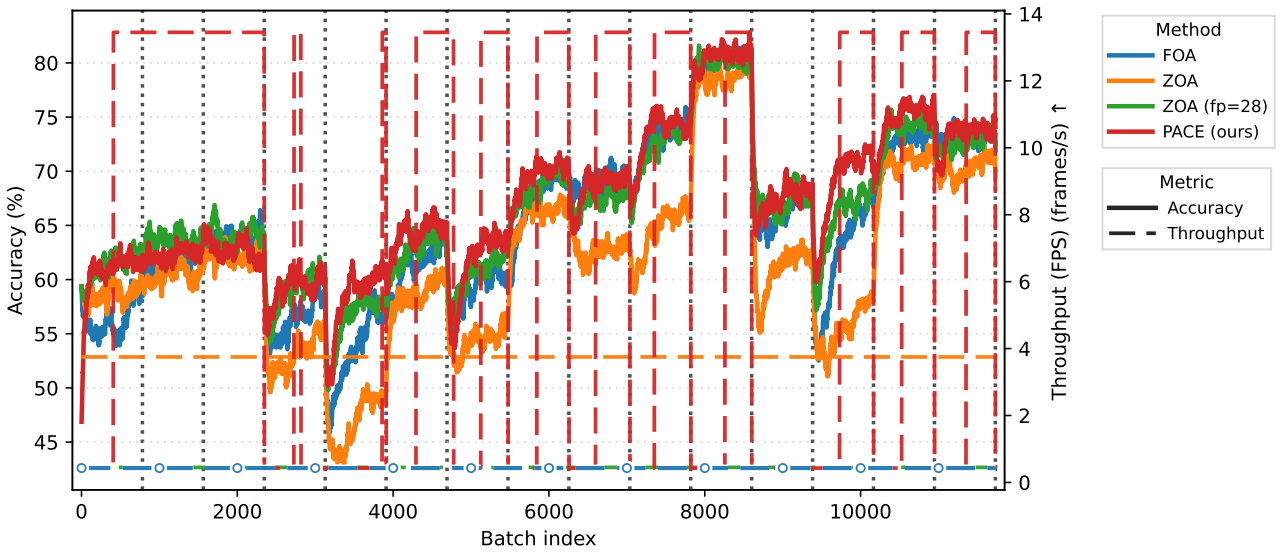

Figure A.2. Smoothed per-batch accuracy for the ImageNet-C benchmark with ViT-B model. The throughput was measured on JETSON XAVIER NX. The gray vertical lines indicate the start of a domain.

Table A.2. Comparisons with SOTA methods on ImageNet-C with ViT-B regarding **Accuracy (%)**. **BP** is short for backward propagation.

		Noise			Blur				Weather				Digital				
Method	BP	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elas.	Pix.	JPEG	Avg.
NoAdapt	✗	56.8	56.8	57.5	46.9	35.6	53.1	44.8	62.2	62.5	65.7	77.7	32.6	46.0	67.0	67.6	55.5
TENT	✔	57.6	59.8	60.9	51.2	49.4	59.6	53.2	64.0	62.7	67.8	78.6	66.5	54.5	70.0	69.7	61.7
CoTTA	✔	57.4	58.4	59.7	47.5	38.3	54.9	47.3	62.4	63.4	69.9	77.8	54.3	47.8	68.0	68.6	58.4
SAR	✔	59.1	61.1	61.6	54.2	55.1	58.6	55.7	60.3	61.5	64.3	76.6	58.2	58.1	68.6	68.9	61.5
LAME	✗	56.5	56.5	57.2	46.4	34.7	52.7	44.2	58.4	61.6	63.1	77.4	24.7	44.6	66.6	67.2	54.1
T3A	✗	56.4	56.6	56.7	45.5	34.4	51.9	43.4	60.6	62.8	60.9	77.1	45.8	44.5	66.7	68.5	55.4
FOA	✗	56.3	61.7	63.9	56.1	53.3	61.4	58.9	67.5	69.1	73.0	80.2	65.9	62.1	72.5	73.0	65.0
ZOA	✗	58.6	60.5	62.3	52.9	46.7	58.8	54.2	66.0	62.7	64.5	78.8	60.6	55.4	70.9	70.3	61.5
ZOA (fp=28)	✗	61.3	63.7	64.1	58.8	56.1	62.8	60.5	68.5	67.4	72.1	80.2	66.9	65.4	73.7	72.6	66.3
PACE (ours)	✗	61.2	62.3	62.7	59.1	58.2	64.2	61.3	69.6	68.4	73.5	80.7	67.1	68.7	74.9	73.3	67.0

Table A.3. Comparisons with state-of-the-art methods on ImageNet-C with ViT-B in long-term continual adaptation. We report average accuracy at each round of continual adaptation. **Bold** indicates best result, underlined second.

Method	round 1	round 2	round 3	round 4	round 5
NoAdapt	55.5	55.5	55.5	55.5	55.5
T3A	55.4	55.9	55.2	55.0	54.6
FOA	65.0	65.6	66.1	66.2	66.4
ZOA	61.5	63.0	63.2	63.9	64.0
ZOA (fp = 28)	66.3	67.4	67.3	67.8	**68.0**
PACE (ours)	**67.0**	**67.9**	**67.8**	**68.1**	67.8

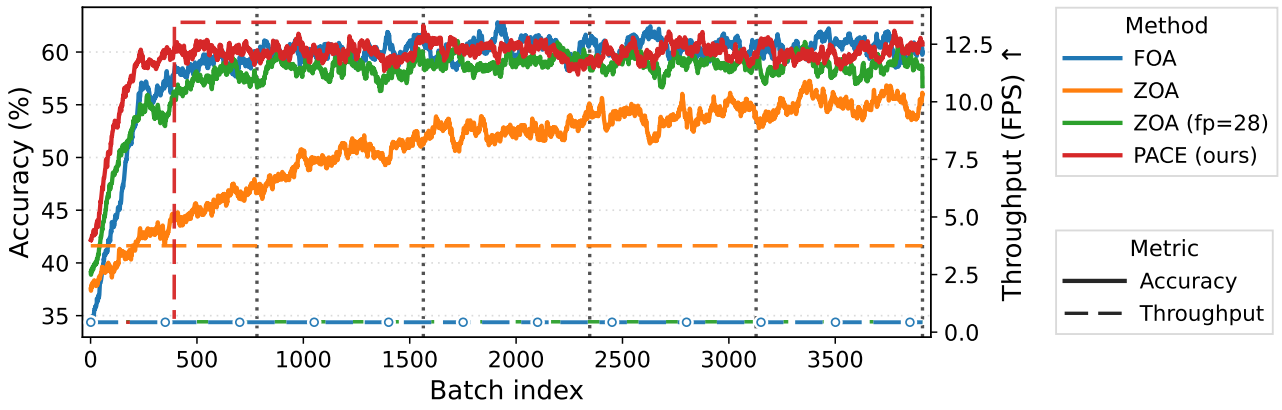

Figure A.3. Smoothed per-batch accuracy for the repeated *Glass Blur* domain from ImageNet-C with ViT-B model. The throughput was measured on a JETSON XAVIER NX. The gray vertical lines indicate the start of each repetition. The throughput from ZOA (fp = 28) is covered by FOA.

PACE achieves a significantly high accuracy at the first round, then peaks by the second and maintains stability across the subsequent ones. Only ZOA (fp = 28) outperforms PACE after five repetitions, gaining a marginal 0.2 percentage points of accuracy after the fourth repetition, while adapting to all test samples. Fig. A.3 illustrates per-batch accuracy and throughput for repeated *Glass Blur* domain from ImageNet-C. PACE identifies when adaptation is no longer necessary and terminates redundant updates, maintaining a throughput of 13 fps on the JETSON XAVIER NX. While FOA and ZOA (fp = 28) maintain similar accuracy, they expend computational resources by adapting throughout the entire sequence, resulting in a significantly lower throughput of 0.45 fps.

A.5.4. Ablation Studies

Optimization dimensionality (d). Tab. A.4 (a) shows performance degradation when d is either increased or decreased. The reduction in accuracy at lower dimensions indicates insufficient expressivity for the optimization updates. Meanwhile, the drop at higher dimensions is likely due to the increased search space complexity, which would require a larger CMA-ES population than our budget of 28.

Domain-Specialized Vector Bank maximum capacity (p). Tab. A.4 (b) shows the impact of the domain-specialized vector bank capacity p on the ImageNet-C benchmark. Performance improves as p increases, reaching a plateau at $p = 15$. This behavior is expected since ImageNet-C consists of 15 distinct domains. As shown in Fig. A.2, our method successfully detects almost every domain transition and initializes from the bank (see Throughput). Consequently, increasing the capacity beyond the number of available domains yields no further gains.

Domain shift threshold γ. The effect of domain shift threshold γ is evaluated in Tab. A.4 (c). While a lower γ increases sensitivity to distribution shifts, frequent resets prevent the CMA-ES from reaching an optimal distribution. On the other hand, a higher γ results in insufficient sensitivity, leaving the model with suboptimal starting points when significant shifts occur. Our results suggest that a balanced threshold is necessary to maintain both detection accuracy and optimization stability.

Table A.4. Accuracy (%) of PACE on ImageNet-C with a ViT-B model.

(a) low-dimension space (d)

	300	768	2304	3000
PACE (ours)	63.8	65.4	**67.0**	66.5

(b) maximum vector bank capacity (p)

	0	5	15	30	40	50
PACE (ours)	66.3	66.6	**67.0**	**67.0**	**67.0**	**67.0**

(c) domain shift threshold (γ)

	0.01	0.02	0.03	0.05	0.1	1.5
PACE (ours)	65.9	**67.0**	**67.0**	66.5	66.6	63.0