EVA-0: Test-Time Model Evolution with Only *Two* Forward Passes per Sample

Guohao Chen[1*] **Shuaicheng Niu**[1,2*] **Geng Li**[1] **Yunbei Zhang**[3] **Shilin Shan**[1]
Chunyan Miao[1] **Jianfei Yang**[1†]
[1]Nanyang Technological University, [2]Jinan-NTU Green Technology Research Institute, [3]Tulane University
cascol-chen.github.io/EVA-0

Abstract

Test-time model evolution offers a promising way for deployed models to improve from unlabeled test-time experience, yet most existing methods depend on back-propagation (BP), which incurs substantial memory overhead and makes them difficult to deploy on edge devices, quantized models, specialized accelerators, or black-box models. In this work, we **study test-time model evolution under a strict two-forward budget**, a setting that pushes adaptation toward highly efficient real-world deployment. We reveal three key obstacles in zeroth-order test-time optimization: susceptibility to shortcut solutions, uncontrolled weight drift, and ineffective update direction estimation. To overcome them, we propose EVA-0, a minimal zeroth-order adaptation framework that: 1) keeps the loss scale-invariant to prevent shortcut solutions; 2) devises an anchor-guided optimization strategy to alleviate weight drift; 3) uses sample-wise symmetric two-sided perturbation for update direction estimation and inference. EVA-0 requires no BP and performs both inference and adaptation within only two forward passes per sample. Results on ImageNet-C&ViT-Base show that EVA-0 outperforms both BP-based DeYO and BP-free FOA, while achieving a 14× speed-up over FOA.

1 Introduction

Next-generation AI is expected to advance not only through larger pre-training, but also by *learning from its own experience* after release [44, 12]. In deployment, such experience is abundant: models constantly encounter data from changing conditions, sensors, and users [25, 29], which are unseen in training and often observed only once. A frozen model simply discards this experience and becomes suboptimal as conditions shift. This calls for test-time model evolution, where a deployed model continually improves itself from its unlabeled test stream and remains stable over long horizons. As such evolution must happen wherever the model runs, *learning should be nearly as cheap as inference*.

Test-Time Adaptation (TTA) [34, 48, 49] is the most widely studied form of this capability: adapting a model online with unlabeled test-time experience under distribution shifts. However, most TTA methods remain difficult to deploy as they rely on backpropagation (BP) to compute gradients [39], which makes each update far more costly than inference, and can be incompatible with resource-limited edge devices, quantized models, specialized accelerators, or black-box model access settings.

Recent works have begun to explore BP-free TTA. FOA [39] conducts forward-only adaptation through prompt evolution with CMA-ES [22]. ZOA [9] extends further by applying zeroth-order (ZO) optimization to quantized networks under a two-forward budget with one-sided SPSA [45]. However, these solutions either require many forward evaluations or produce less effective update directions under a strict two-forward budget, resulting in low efficiency or limited performance. Thus, it remains unclear whether a deployed model can *stably and effectively evolve from unlabeled test-time experience using only two forward passes per sample?*

*Equal contributions. †Corresponding author. Emails: {guohao.chen, shuaicheng.niu, jianfei.yang}@ntu.edu.sg

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To answer this question, we revisit ZO test-time optimization under the two-forward constraint through analytical and empirical studies, revealing both its key challenges and discovering new opportunities for advanced methodology design. From Figures 1–2, we have following key takeaways:

• ZO optimization is more prone to learn shortcut solutions than backpropagation (BP) under test-time loss functions, while keeping the loss scale-invariant can help mitigate this issue (c.f. Sec. 3.1).

• Unlike BP, random ZO updates tend to cause weight drift. By properly controlling the optimization trajectory to alleviate such drift, ZO test-time learning can be better stabilized (c.f. Sec. 3.2).

• *Two* forward passes are far more informative than they appear. Averaging two symmetrically perturbed predictions approximates the original prediction, so one can save the extra pass for inference and use both for a more reliable two-sided update. Crucially, a separate perturbation per sample improves the diversity of update directions and their alignment with BP gradients (c.f. Sec. 3.3).

Inspired by the above analyses, we propose **EVA-0**, short for **Ev**olutionary **A**daptation with **Zero**th-Order gradients, a minimal framework for stable test-time model evolution under the strict two-forward constraint. EVA-0 is designed from three principles: **1) shortcut-resistant optimization:** EVA-0 employs *scale-invariant test-time objectives*, which prevent ZO updates from reducing entropy through logit-norm inflation and encourage meaningful test-time learning; **2) controlled optimization trajectory:** EVA-0 introduces *anchor-guided optimization*, which mitigates drift by periodically steering exploration relative to a reliable anchor, thereby enabling stable and continual model evolution; **3) accurate and informative update direction estimation:** EVA-0 uses symmetric *two-sided SPSA* to obtain lower-variance update signals while averaging perturbed outputs for inference. It further proposes *sample-wise independent perturbation* to enhance the diversity of explored update directions that significantly improves the alignment with BP update. All these mechanisms operate within the same two forward passes per sample, and EVA-0 requires no backpropagation. As a result, EVA-0 enables effective, efficient, and stable model evolution from unlabeled test-time experience.

Main contributions 1) We study test-time model evolution under a strict two-forward budget, a practical and challenging setting that pushes test-time learning/adaptation toward highly efficient real-world deployment at scale. 2) We systematically analyze the limitations of current ZO test-time optimization, revealing key challenges and design opportunities in shortcut exploitation, weight drift, and limited or biased direction estimation. 3) We propose EVA-0, a two-forward test-time evolution framework that addresses these challenges with scale-invariant shortcut-resistant optimization, anchor-guided trajectory control, and sample-wise two-sided perturbation sampling and inference. 4) We validate EVA-0 across CNNs, ViTs, quantized models, continual streams, and black-box prompt adaptation, demonstrating effective model evolution with a practical accuracy-efficiency trade-off.

2 Preliminary and Problem Statement

Test-Time Adaptation (TTA) Let $f_{\theta_0} : \mathcal{X} \to \mathbb{R}^C$ be a source model trained on labeled source data. During deployment, the model receives unlabeled test mini-batches $\{\mathcal{B}_t\}_{t=1}^T$ from a potentially shifted target distribution. Online TTA updates f_{θ_0} by optimizing an unsupervised test-time objective:

$$\theta_{t+1} = \theta_t - \eta \hat{g}_t, \qquad \hat{g}_t \approx \nabla_\theta \mathcal{L}(\theta_t; \mathcal{B}_t), \tag{1}$$

where $\mathcal{L}$ can be entropy minimization [48] or consistency [56], *etc.* Most prior methods obtain $\hat{g}_t$ via backpropagation, which might be impractical in resource-constrained or non-differentiable settings.

Test-Time Forward Optimization has been recently proposed to avoid BP. At each step, the learner evaluates perturbed models $\mathcal{Q}_t = \{f_{\theta_t+\delta_k}(\mathcal{B}_t)\}_{k=1}^K$, computes the corresponding unsupervised losses, and updates θ_t only from these forward signals. FOA [39] pioneers this direction by using CMA-ES [22] to evolve input prompts. However, CMA-ES typically requires many forward evaluations, *e.g.*, $K = 28$, and is mainly suitable for low-dimensional prompt spaces due to its high covariance estimation cost, limiting its scalability. ZOA [9] adopts one-sided SPSA [45] for this goal:

$$\widehat{g}_t^{\text{1s}} = \frac{\ell(\theta_t + \mu z_t; \mathcal{B}_t) - \ell(\theta_t; \mathcal{B}_t)}{\mu} z_t, \qquad z_t \sim \mathcal{D}, \tag{2}$$

where z_t is a random perturbation and μ is the perturbation scale. Although ZOA enables updates in higher-dimensional parameter spaces, its estimated update direction is still coarse and noisy, leaving a substantial room for performance improvement. Moreover, both FOA and ZOA rely on feature alignment,which requires source statistics that may be unavailable due to concerns such as privacy.

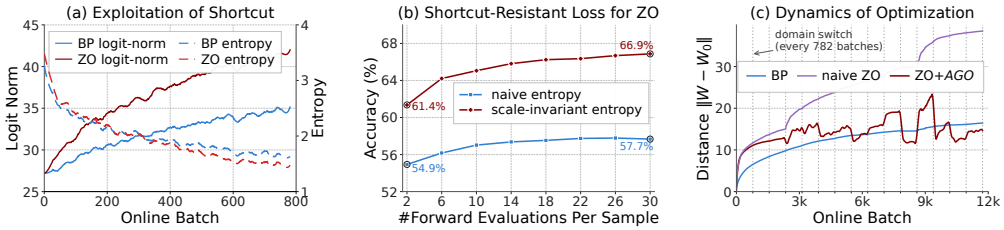


Figure 1: (a) ZO optimization is more vulnerable than BP to shortcut learning: reducing entropy via logit-norm inflation. (b) Scale-invariant objective mitigates the shortcut issue and boosts performance. (c) ZO updates suffer from weight drift, and anchor-guided optimization (AGO) is able to control such drift. Results obtained on ViT-Base&ImageNet-C, 15 corruptions for (b-c) and Gaussian for (a).

Two-Forward Test-Time Optimization (TTO) We study *test-time model evolution* under a strict two-forward constraint, to ensure high run-time and memory efficiency for real-world deployment. At each online step, the model receives an unlabeled mini-batch $\mathcal{B}_t$ (the batch size could be 1) and is allowed only two forward passes per sample, without backpropagation. We build our method on SPSA and address the limitations of prior one-sided forward-only adaptation.

3 A Closer Look at Two-Forward TTO: Challenges and Opportunities

3.1 On the Learning Objective

Challenge: **Zeroth-order (ZO) optimization is more prone than backpropagation (BP) to learning shortcut solutions with test-time learning objectives.** Existing forward-only TTA methods often reuse objectives designed for BP-based adaptation in ZO updates, assuming BP-effective losses also work well for ZO. However, we show that BP and ZO differ in their sensitivity to shortcut solutions, where the test-time objective is naively reduced without improving task-relevant generalization. In particular, ZO is more susceptible to such shortcuts than BP, as formalized by

Proposition 1 (***Vulnerability of ZO to Shortcut Signals***)**.** *Let e_s be a unit shortcut direction and decompose the local gradient as $g = Ae_s + g_m$, where $g_m \perp e_s$. Consider the Gaussian ZO estimator $\hat{g}_{\mathrm{ZO}} = (g^\top z + \xi)z$, with $z \sim \mathcal{N}(0, I)$ and zero-mean loss noise ξ. For any unit direction $v \perp e_s$,*

$$\mathbb{E}\big[v^\top \hat{g}_{\mathrm{ZO}}\big] = v^\top g_m, \qquad \mathrm{Var}\big(v^\top \hat{g}_{\mathrm{ZO}}\big) \geq A^2.$$

Indeed, $v^\top \hat{g}_{\mathrm{ZO}} = A(e_s^\top z)(v^\top z) + (g_m^\top z)(v^\top z) + \xi(v^\top z)$. The first term is induced solely by the shortcut signal, which has zero mean but variance A^2. Thus, the shortcut signal injects large variance into orthogonal directions related to meaningful updates, and may even flip meaningful updates. In contrast, BP directly accesses $v^\top g = v^\top g_m$ without mixing shortcut signals into orthogonal directions.

Opportunity: **Keeping the entropy objective scale-invariant could help mitigate shortcut learning.** We further empirically study the above proposition using the widely used entropy loss [48, 40, 30]. Entropy can be reduced either by improving the decision boundary or by simply inflating the logit norm, the latter forming a shortcut direction without meaningful learning. As in Figure 1 (a), when BP and ZO achieve similar entropy reduction, ZO induces much larger logit norms, indicating its stronger susceptibility to such shortcuts (details in App. E). Moreover, as in Figure 1 (b), making entropy logit-norm-invariant (*i.e.*, Eqn. (6) without de-centering) significantly improves ZO performance. Thus, two-forward model evolution requires objectives that are not only effective but also shortcut-resistant.

3.2 On the Weight Dynamics

Challenge and Opportunity: **Random ZO updates can accumulate into weight drift, whereas BP updates are less prone to this issue.** Unlike BP, ZO updates rely on random perturbation directions scaled by loss differences. Thus, even when a step reduces the current test-time loss, it may include components unrelated to stable model improvement, especially under noisy unsupervised objectives. These components can accumulate over online streams and move the model away from the pretrained stable region. Since later perturbations are independently sampled, they are unlikely to align with the recovery direction in high-dimensional space. As in Figure 1 (a), naive ZO lead to growing weight norms and degraded performance (Table 4), whereas BP-based updates remain more stable under the same objective. This motivates trajectory control in addition to objective minimization.

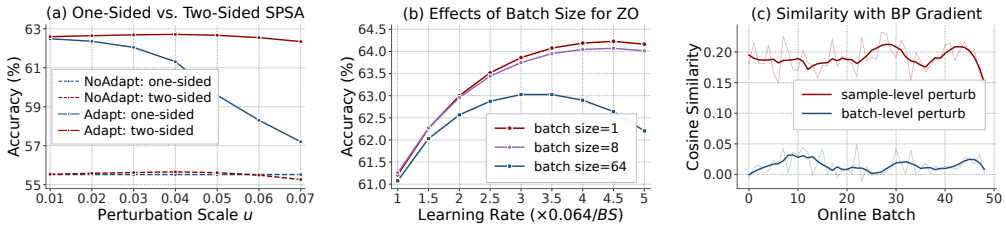


Figure 2: (a) Accuracy of different direction estimation and inference strategies. *One-sided* SPSA uses the original output for inference, while *two-sided* SPSA averages two perturbed outputs to approximate clean inference. (b) Smaller batches provide more independent perturbations, yielding more diverse update directions and better performance. (c) Sample-wise perturbations yield ZO directions better aligned with BP gradients. Results obtained on ViT-Base & ImageNet-C (15 corruptions for (a-b) and Gaussian for (c)) using CE loss.

3.3 On the Gradient Direction Estimation Strategy

Challenge: **One-sided SPSA is sensitive to perturbation noise, while two-sided SPSA requires more forward passes.** Existing forward-only TTA methods typically need to allocate one forward pass to the original model for inference. Thus, under a strict two-forward budget, one has only a single perturbed evaluation (one forward pass) for estimating the update direction, as one-sided SPSA in Eqn. (2). However, as in Figure 2 (a), one-sided SPSA preserves stable inference but makes adaptation effectiveness limited and sensitive to perturbation noise. In contrast, two-sided SPSA uses the two evaluations as a symmetric local probe around the current model:

$$\widehat{g}_t^{\mathrm{2s}} = \frac{\ell(\theta_t + \mu z_t; \mathcal{B}_t) - \ell(\theta_t - \mu z_t; \mathcal{B}_t)}{2\mu} z_t, \qquad z_t \sim \mathcal{D}. \tag{3}$$

This usually yields a more reliable update direction, but it appears to require an additional clean forward pass for inference, *i.e.*, at least three forward passes, violating the two-forward constraint.

Opportunity: **Averaging two-sided probing outputs approximates clean-model inference without an additional forward pass.** To ensure two-forward efficiency, unlike standard uses of two-sided SPSA merely as a variance-reduction estimator, we further exploit its symmetric outputs for inference. Under a local smoothness assumption, a Taylor expansion around θ_t gives

$$f_{\theta_t \pm \mu z_t}(x) = f_{\theta_t}(x) \pm \mu J_{\theta_t}(x) z_t + O(\mu^2). \tag{4}$$

Thus, averaging $f_{\theta_t + \mu z_t}(x)$ and $f_{\theta_t - \mu z_t}(x)$ cancels the first-order term and yields a second-order approximation to clean-model inference. This reveals a useful opportunity for two-forward test-time evolution: the same two symmetric evaluations can provide both a lower-variance update direction and a reliable inference output, without requiring an additional clean forward pass. Empirically, Figure 2 (a) confirms that averaged symmetric predictions remain stable across perturbation scales, while the resulting update signals lead to consistently stronger adaptation.

Opportunity: **Sample-level perturbations improve directional diversity than shared batch-level perturbations.** Applying a *shared perturbation* across a mini-batch is a common practice in ZO optimization, as in Eqns. (2-3). Recent ZO studies further suggest using *large batches* to reduce loss variance and stabilize offline training [17]. However, under two-forward constraint, a mini-batch with a shared perturbation explores only one direction in parameter space, regardless of the batch size.

In the online learning setting, where each sample is used only *once*, we observe a phenomenon that contrasts with prior studies: smaller batches often lead to better adaptation. As in Figure 2 (b), reducing the batch size consistently improves performance across learning rates. This suggests that, in streaming ZO TTO, *directional exploration can be more valuable than batch-level loss averaging*. To directly measure this effect, we use the backpropagation gradient on the same mini-batch as an oracle reference and compare its similarity to the zeroth-order update direction. As in Figure 2 (c), *independent perturbations* across samples yield substantially higher gradient similarity than a shared batch-level perturbation, *i.e.*, increasing similarity with BP gradient from *0.015* to *0.190*. This provides direct evidence that test samples can contribute useful directional information beyond scalar loss averaging, motivating sample-wise directional exploration under the two-forward constraint.

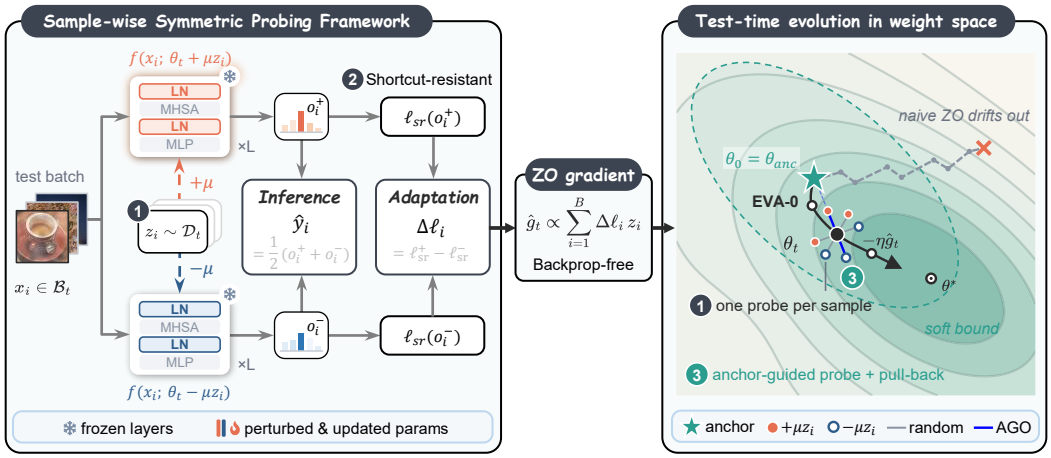


Figure 3: An illustration of our proposed EVA-0. For each test sample x_i, we draw an independent perturbation $z_i \sim \mathcal{D}_t$ and run two symmetric forward passes $f(x_i; \theta_t \pm \mu z_i)$ (①), whose averaged output gives the prediction and whose shortcut-resistant loss difference $\Delta\ell_i$ (②) yields a backpropagation-free gradient $\hat{g}_t$. We then update θ_t under anchor-guided optimization (③), so that EVA-0 evolves toward the optimum within a soft bound, while naive ZO updates drift away.

4 EVA-0: Stable and Effective Two-Forward Model Evolution at Test Time

In this section, we propose **EVA-0**, short for **Ev**olutionary **A**daptation with **Zero**th-order gradients. EVA-0 aims to achieve stable and effective two-forward test-time model evolution by addressing the challenges revisited in Sec. 3. Motivated by the opportunities discovered in Sec. 3, EVA-0 instantiates them as three design principles: 1) sample-wise two-sided perturbation for direction estimation and inference (c.f. Sec. 4.1); 2) shortcut-resistant objective (c.f. Sec. 4.2); 3) bounded yet plastic online optimization (c.f. Sec. 4.3), and then integrates them into a simple, unified 2-forward ZO update rule:

$$\hat{g}_t^{\text{EVA}} = \frac{1}{B}\sum_{i=1}^{B} \frac{\ell_{\text{sr}}(\theta_t + \mu z_{t,i}; x_i) - \ell_{\text{sr}}(\theta_t - \mu z_{t,i}; x_i)}{2\mu} z_{t,i}, \qquad z_{t,i} \sim \mathcal{D}_t, \tag{5}$$

Here, ℓ_{sr} denotes a *shortcut-resistant* test-time objective to encourage meaningful learning, and $\mathcal{D}_t$ is an adaptive rather than fixed perturbation distribution that co-evolves with θ_t to control the model trajectory. The overall **pseudo-code of EVA-0 is summarized in Algorithm 1 in Appendix A.**

4.1 Sample-wise Symmetric Update Direction Estimation and Inference ①

Following Sec. 3.3, EVA-0 uses *sample-wise symmetric direction estimation* (SSD). For each sample x_i, we draw an independent perturbation $z_{t,i} \sim \mathcal{D}_t$ and evaluate two symmetric models:

$$f_{\theta_t + \mu z_{t,i}}(x_i) \quad \text{and} \quad f_{\theta_t - \mu z_{t,i}}(x_i).$$

The two forward passes are used for both inference and adaptation. **For adaptation**, EVA-0 uses the two-sided loss difference, as defined in Eqn.(5). **For inference**, EVA-0 averages the two perturbed outputs, $f_{\theta_t \pm \mu z_{t,i}}(x_i)$, which provides a second-order approximation to the clean output under local smoothness, as shown in Figure 2 (a).

This symmetric strategy satisfies the two-forward constraint while retaining the more reliable direction estimation enabled by two-sided perturbation. The sample-wise perturbations further turn each mini-batch into *multiple directional probes*, rather than a single loss-averaging unit. As a result, EVA-0 obtains update directions that are both more reliable and more informative from streaming test samples. We provide a memory- and computation-efficient implementation of SSD in Appendix C.1.

4.2 Shortcut-Resistant and Sample-Wise Test-Time Objectives ②

As discussed in Sec. 3.1, ZO optimization is more prone than BP to learning shortcut solutions, and maintaining a scale-invariant objective can help mitigate this issue. Therefore, in this section, we

introduce two scale-invariant objectives for two-forward TTO. The first works in the fully source-free TTA setting, while the latter can further improve performance when source statistics are available.

Shortcut-resistant entropy minimization Vanilla entropy minimization is effective for TTA, but it admits shortcut solutions, which become more severe from BP to ZO optimization: the loss can be reduced by inflating logit norms or collapsing predictions to a few high-confidence classes. EVA-0 mitigates these shortcuts via scale normalization and output decentering. Given symmetric logits $o_{t,i}^{\pm} = f_{\theta_t \pm \mu z_{t,i}}(x_i)$, let $\bar{o}_{t,i} = (o_{t,i}^{+} + o_{t,i}^{-})/2$ and $\bar{r}_{t,i} = \|\bar{o}_{t,i}\|_2$. For $o \in \{o_{t,i}^{+}, o_{t,i}^{-}\}$, we define

$$s(o) = \frac{\bar{r}_{t,i}}{\|o\|_2 + \epsilon} o - c_t, \qquad p(o) = \mathrm{softmax}(s(o)), \qquad E^{\mathrm{SR}}(o) = -\sum_k p_k(o) \log p_k(o), \quad (6)$$

where c_t is an online output center maintained via EMA over $\bar{o}_{t,i}$, inspired by DINO [4]. The normalization term removes the logit-scale degree of freedom by evaluating both perturbed outputs at the same norm $\bar{r}_{t,i}$, preventing entropy reduction through norm inflation. The decentering term improves output diversity by suppressing persistent output collapse. Together, they allow ZO to extract more meaningful updates from entropy.

Sample-wise feature-statistic alignment When source feature statistics are available, feature alignment provides an additional reference for meaningful adaptation, as in FOA [39] and ZOA [9]. The alignment loss is naturally more scale-invariant. However, under sample-wise perturbations (as described in Sec. 4.1) and single-sample adaptation scenarios, it becomes ill-posed because target statistics are estimated from a single sample, yielding noisy means and ill-defined variances.

EVA-0 makes this alignment well-defined by estimating how each sample updates the online target distribution. As one sample has no variance, EVA-0 instead tracks the first and second moments over the test stream and recovers the variance via $\mathrm{Var}[h] = \mathbb{E}[h^2] - \mathbb{E}[h]^2$. Formally, let $h_\theta(x)$ be the alignment feature with source mean and standard deviation (m_s, σ_s), and (m_t, q_t) be the running estimates of $\mathbb{E}[h]$ and $\mathbb{E}[h^2]$. For a perturbed feature $h = h_{\theta_t \pm \mu z_{t,i}}(x_i)$, the updated moments are

$$\hat{q}_t(h) = (1-\rho) q_t + \rho h^2, \qquad \hat{m}_t(h) = (1-\rho) m_t + \rho h, \qquad \hat{\sigma}_t(h) = \sqrt{\hat{q}_t(h) - \hat{m}_t(h)^2}. \quad (7)$$

The sample-wise alignment loss is then computed by

$$\ell_{\mathrm{SWA}}(h) = \|\hat{m}_t(h) - m_s\|_2^2 + \|\hat{\sigma}_t(h) - \sigma_s\|_2^2. \quad (8)$$

This EMA formulation makes feature alignment compatible with sample-wise two-forward evolution: each sample provides a valid alignment signal, while moving moments stabilize the online estimates. After each update, (m_t, q_t) are updated using the averaged symmetric features, allowing the statistics to evolve online without extra forward passes. Then, the **Overall Objective** of EVA-0 is given by

$$\ell_{\mathrm{sr}}(\theta; x) = E^{\mathrm{SR}}(o) + \lambda \ell_{\mathrm{SWA}}(h). \quad (9)$$

When source feature statistics are unavailable, we set $\lambda = 0$, yielding fully test-time adaptation.

4.3 Anchor-Guided Bounded Test-Time Evolution ③

Following Sec. 3.2, EVA-0 achieves optimization trajectory control with an adaptive perturbation distribution $\mathcal{D}_t$. Let θ^{anc} be a reliable teacher or anchor, and $d_t = \theta^{\mathrm{anc}} - \theta_t$ denote the anchor direction. For each mini-batch $\mathcal{B}_t$ with B samples, EVA-0 assigns $B - k$ samples to random exploration ($i \notin \mathcal{I}_t$) and remaining k samples $i \in \mathcal{I}_t \subset \mathcal{B}_t$ to anchor-guided perturbations. That is,

$$z_{t,i} \sim \mathcal{D}_t = \begin{cases} \bar{d}_t \odot r_{t,i}, & i \in \mathcal{I}_t, \quad r_{t,i} \sim \mathcal{U}(0,2), \\ u_{t,i}, & i \notin \mathcal{I}_t, \quad u_{t,i} \sim \mathcal{N}(0, I), \end{cases} \quad \text{where } \bar{d}_t = \frac{\mathbb{E}_{u \sim \mathcal{N}(0,I)}[\|u\|_2]}{\|d_t\|_2 + \delta} d_t. \quad (10)$$

Here, $\odot$ is element-wise multiplication. Under the symmetric two-forward estimator, the two sides $\theta_t + \mu z_{t,i}$ and $\theta_t - \mu z_{t,i}$ correspond to moving relatively closer to or farther from the anchor. The shortcut-resistant loss compares these two directions and determines which one better fits the current test sample. As a result, anchor-guided perturbation provides a recovery-aware search direction that can pull the model back toward a stable region when needed, while still allowing adaptation away from the anchor when supported by test-time evidence.

The above anchor-guided perturbation controls the update direction in a loss-driven manner. Here, we further apply a weak relaxation that acts as a smoother to suppress the parameter drift:

$$\theta_{t+1} = (1-\gamma)\theta'_{t+1} + \gamma \theta^{\mathrm{anc}}, \quad (11)$$

Table 1: Comparisons with SOTA methods on ImageNet-C (severity level 5) regarding **Accuracy (%)** under `single-domain TTA`, **BP** is short for backward propagation, **FTTA** is short for source-free fully TTA, and **#FP** is the number of forward passes a sample. † indicates a variant that uses only prediction entropy as the test-time objective, *i.e.*, λ=0. The **bold** number indicates the best result.

					Noise			Blur				Weather				Digital				
	Model+Method	BP	FTTA	#FP	Gaus.	Shot	Imp.	Def.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elas.	Pix.	JPEG	Avg.
ViT-Base	NoAdapt	✗	✔	1	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	✔	✔	1	60.3	61.6	61.8	59.2	56.5	63.5	59.2	54.3	64.5	2.3	79.1	67.4	61.5	72.5	70.6	59.6
	CoTTA	✔	✔	3or35	63.6	63.8	64.1	55.5	51.1	63.6	55.5	70.0	69.4	71.5	78.5	9.7	64.5	73.4	71.2	61.7
	SAR	✔	✔	[1, 2]	59.2	60.5	60.7	57.5	55.6	61.8	57.6	65.9	63.5	69.1	78.7	45.7	62.4	71.9	70.3	62.7
	DeYO	✔	✔	[1, 2]	59.8	61.0	60.2	58.4	58.8	64.4	62.3	68.8	66.4	70.1	78.8	62.2	68.4	74.1	72.2	65.7
	ZOA	✗	✗	2	58.2	59.7	60.3	52.6	47.2	58.7	53.9	66.1	62.9	67.6	78.7	56.9	56.8	70.2	70.1	61.3
	FOZO	✗	✗	2	58.7	61.1	62.5	51.1	44.7	57.8	53.4	66.3	66.3	67.8	79.1	59.8	55.4	70.4	71.8	61.7
	FOA	✗	✗	28	61.5	63.2	63.3	59.3	56.7	61.4	57.7	69.4	69.6	73.4	81.1	67.7	62.7	73.9	73.0	66.3
	FOA†	✗	✔	28	58.2	59.6	59.1	55.1	36.0	54.0	47.3	47.8	59.1	72.5	77.2	30.0	48.7	66.6	67.9	55.9
	EVA-0 (ours)	✗	✗	2	61.8	63.0	63.8	58.1	56.4	62.3	60.5	70.0	68.5	74.2	80.7	68.5	66.4	73.8	73.2	**66.7**
	EVA-0† (ours)	✗	✔	2	60.1	61.3	62.0	56.8	54.7	60.8	57.4	67.7	66.2	71.7	79.4	66.7	62.1	72.0	70.3	64.6
R50-GN	NoAdapt	✗	✔	1	22.1	23.0	22.0	19.8	11.4	21.5	25.0	40.3	47.0	34.0	68.8	36.2	18.5	29.2	52.6	31.4
	TENT	✔	✔	1	24.2	28.0	25.0	15.2	7.7	22.0	22.6	26.9	33.9	3.5	69.9	42.1	11.0	48.6	54.4	29.0
	CoTTA	✔	✔	3or35	32.3	34.2	34.6	12.4	15.9	13.9	26.4	39.8	39.6	7.2	66.7	23.9	18.1	50.2	56.8	31.5
	SAR	✔	✔	[1, 2]	32.3	34.6	33.5	18.9	19.7	30.4	30.8	42.6	43.3	6.2	70.4	43.8	15.9	49.0	55.2	35.1
	DeYO	✔	✔	[1, 2]	40.7	42.7	42.3	21.6	23.6	38.1	36.4	51.1	49.4	54.9	72.9	50.1	42.1	56.3	57.7	45.3
	ZOA	✗	✗	2	29.0	29.2	27.7	23.6	15.2	27.3	30.2	44.5	46.5	49.9	69.9	42.4	26.3	41.6	55.8	37.3
	FOZO	✗	✗	2	32.1	32.5	33.7	13.1	8.6	18.0	21.3	38.8	44.3	31.4	68.0	35.2	22.8	51.4	55.3	33.8
	FOA	✗	✗	28	27.3	27.6	23.1	21.1	11.5	23.7	26.9	38.6	48.5	42.0	70.1	41.6	20.7	51.4	55.3	35.3
	FOA†	✗	✔	28	26.9	27.0	20.7	21.0	11.2	23.3	27.0	39.3	48.1	42.4	70.0	41.7	21.4	51.3	55.0	35.1
	EVA-0 (ours)	✗	✗	2	37.0	38.2	37.9	29.1	24.9	37.2	41.5	51.0	51.0	59.5	72.6	47.9	41.7	54.2	57.3	**45.4**
	EVA-0† (ours)	✗	✔	2	32.3	35.2	34.2	23.8	15.5	32.6	34.9	47.4	46.9	52.6	71.4	44.8	31.4	51.8	56.2	40.7

where θ'_{t+1} is an intermediate model after the ZO update. With a small γ, the model remains plastic enough to learn from test-time evidence, while its trajectory is denoised and softly bounded that prevents drift. In EVA-0, θ^{anc} can be realized as the pretrained weights or a slowly moving EMA of the online parameters (see Table 7).

5 Comparisons with State-of-the-arts

We organize the experiments to demonstrate the efficacy of EVA-0 across *CNNs*, *ViTs*, *continual streams*, *quantized models*, *black-box models*, and provide further insights into the designs of EVA-0.

Dataset and methods We conduct experiments on ImageNet-C [25], a large-scaled benchmark to assess out-of-distribution generalization. It contains 15 types of 4 main categories (noise, blur, weather, digital) corrupted images and each type has 5 severity levels. We benchmark EVA-0 against the following TTO methods: (i) BP-based methods: TENT [48], CoTTA [49], SAR [40], DeYO [30], which use EMA-teacher regularization, sharpness-aware optimization, and active learning for BP-based TTO; (ii) Forward-Only methods: FOA [39], ZOA [9], FOZO [50], that are based on ZO.

Models and implementation details We use both transformer and CNN models, ViT-Base [10] and ResNet50-GN [24], for our experiments. The models are trained on ImageNet-1K and the model weights are obtained from the `timm` repository [51]. We adopt PTQ4ViT [54] for 8-bit and 6-bit ViT quantization. We update the affine parameters in norm layers per ZOA [9], using SGD with a learning rate of 0.002. We use the pretrained weights as θ^{anc} with $\gamma = 0.001$ in Eqn. (11). Discussions on hyperparameters can be found in Sec. 5.2 and Appendix E. More details are put in Appendix D.

5.1 Main Results on Full Precision Models and Quantized Models

Single-domain test-time optimization We first assess the effectiveness of EVA-0 in short-term TTO. Here, we conduct experiments under the single-domain TTA protocol, where the model is reset before evolving on each new target domain. From Table 1, we have the following observations. 1) EVA-0 achieves the best average accuracy on both ViT-Base and ResNet50-GN under a strict two-forward budget without backpropagation. On ViT-Base, EVA-0 outperforms the strong BP-based method DeYO by 1.0%, two-forward ZOA by 5.4%, and 28-forward FOA by 0.4% w.r.t. the average accuracy, suggesting its effectiveness. 2) The advantage of EVA-0 becomes more pronounced on ResNet50-GN, outperforming FOA by 10.1%, which suggests that EVA-0 is not tied to a specific architecture. 3) Importantly, EVA-0† demonstrates the first practical *forward-only fully TTA method*,

Table 2: Comparisons with SOTA methods on ImageNet-C (severity level 5) regarding **Accuracy (%)** under `continuous adaptation`. **BP** is short for backward propagation, **FTTA** is short for source-free fully TTA, and **#FP** denotes the number of forward passes a sample. † indicates a variant that uses only entropy as the test-time objective, *i.e.*, λ=0. The **bold** number indicates the best result.

					Noise			Blur				Weather				Digital				
	Model+Method	BP	FTTA	#FP	Gaus.	Shot	Imp.	Def.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elas.	Pix.	JPEG	Avg.
ViT-Base	NoAdapt	✗	✔	1	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	✔	✔	1	60.3	63.8	64.8	55.0	55.7	60.9	56.4	61.8	63.4	70.0	78.6	64.5	50.1	70.2	70.5	63.1
	CoTTA	✔	✔	3or35	63.4	64.3	61.6	35.9	50.1	47.7	44.6	38.2	48.8	42.3	54.2	5.2	55.5	56.3	54.7	48.2
	SAR	✔	✔	[1, 2]	59.2	61.3	61.6	54.6	55.4	58.8	56.1	61.1	62.2	64.9	76.8	60.3	58.3	68.7	69.2	61.9
	DeYO	✔	✔	[1, 2]	59.8	61.4	61.1	49.8	55.0	55.9	18.2	61.0	61.1	65.2	76.9	49.1	63.8	70.4	69.3	58.5
	ZOA	✗	✗	2	58.2	60.5	61.7	49.9	46.9	57.6	54.4	64.3	62.8	66.3	78.5	59.6	55.7	69.5	70.6	61.1
	FOZO	✗	✗	2	58.7	59.3	60.0	53.9	42.3	57.2	50.6	66.3	61.5	68.5	78.5	62.3	53.6	69.2	70.0	60.8
	FOA	✗	✗	28	61.9	64.0	64.5	57.0	56.7	61.6	61.2	68.8	71.7	74.2	81.0	67.2	61.4	74.1	73.3	66.6
	FOA†	✗	✔	28	58.2	60.0	61.1	40.1	10.5	11.5	12.1	12.7	9.9	16.4	16.9	6.6	5.2	5.1	7.5	22.3
	EVA-0 (ours)	✗	✗	2	61.8	63.6	64.2	58.3	56.3	62.3	60.9	70.1	69.2	74.1	81.0	68.9	66.6	74.0	73.3	**67.0**
	EVA-0† (ours)	✗	✔	2	60.1	62.0	62.6	55.8	53.3	61.2	57.8	67.3	66.0	70.9	79.6	66.8	62.0	72.1	72.1	64.6
R50-GN	NoAdapt	✗	✔	1	22.1	23.0	22.0	19.8	11.4	21.5	25.0	40.3	47.0	34.0	68.8	36.2	18.5	29.2	52.6	31.4
	TENT	✔	✔	1	24.2	6.3	0.8	6.8	1.6	0.9	1.2	1.7	0.4	0.5	5.2	1.0	0.1	0.3	2.2	3.5
	CoTTA	✔	✔	3or35	32.3	41.9	44.9	15.9	28.2	29.2	30.9	24.1	33.0	37.9	49.5	33.4	27.0	46.4	41.3	34.4
	SAR	✔	✔	[1, 2]	32.1	38.7	38.7	27.7	30.5	35.7	42.9	36.7	36.7	47.7	59.3	32.2	47.5	50.8	47.5	40.3
	DeYO	✔	✔	[1, 2]	40.7	48.4	47.9	9.2	12.1	16.3	18.5	3.1	0.8	0.3	0.8	0.2	0.1	0.1	0.1	13.2
	ZOA	✗	✗	2	29.0	27.6	27.8	24.2	14.0	25.0	33.3	42.0	42.3	53.8	70.0	43.7	26.8	34.4	54.1	36.5
	FOZO	✗	✗	2	32.1	34.1	34.2	14.0	6.9	16.6	21.7	25.3	38.8	37.7	56.8	30.9	15.0	49.1	52.2	31.0
	FOA	✗	✗	28	27.3	31.0	30.1	18.4	13.3	23.8	26.7	40.4	25.1	2.3	9.5	1.4	0.5	1.2	1.6	16.8
	FOA†	✗	✔	28	26.8	32.6	31.0	18.5	13.8	24.2	25.7	39.2	12.7	1.9	9.6	2.6	1.4	2.7	3.7	16.4
	EVA-0 (ours)	✗	✗	2	37.0	40.4	39.5	28.2	23.1	38.4	45.1	50.3	51.8	60.2	72.3	49.1	40.5	52.6	56.0	**45.6**
	EVA-0† (ours)	✗	✔	2	32.3	38.4	37.5	24.3	18.8	33.5	38.2	45.3	47.8	53.0	71.4	47.3	29.3	50.6	56.6	41.6

Table 3: Efficacy of our EVA-0 on **Quantized ViT models**. We report **Accuracy (%)** on ImageNet-C (level 5) under `single-domain TTA`. † indicates a variant that uses only entropy as the objective.

				Noise			Blur				Weather				Digital				
	Model+Method	FTTA	#FP	Gaus.	Shot	Imp.	Def.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elas.	Pix.	JPEG	Avg.
8-bit	NoAdapt	✔	1	55.8	55.8	56.5	46.7	34.7	52.1	42.5	60.8	61.4	66.7	76.9	24.6	44.7	65.8	66.7	54.1
	ZOA	✗	2	57.4	57.0	58.2	50.6	45.5	57.6	51.6	64.1	61.2	64.4	78.0	47.5	53.7	68.8	69.1	59.0
	FOZO	✗	2	56.4	56.1	56.9	49.8	38.7	54.4	47.2	64.3	65.3	68.9	77.7	54.6	50.5	67.1	68.4	58.4
	FOA	✗	28	60.7	61.4	61.3	57.2	51.5	59.4	51.3	68.0	67.3	72.4	80.3	63.2	57.0	72.0	69.8	63.5
	FOA†	✔	28	57.1	57.7	58.2	52.1	35.6	54.5	35.3	46.0	48.6	38.3	76.7	21.5	47.6	65.5	66.6	50.8
	EVA-0 (ours)	✗	2	60.9	62.1	62.9	56.5	54.2	61.5	59.2	69.4	68.2	73.2	80.2	63.7	65.8	73.3	72.6	**65.6**
	EVA-0† (ours)	✔	2	59.1	60.4	61.0	56.2	53.3	60.4	56.8	67.4	66.4	71.4	79.4	62.2	62.6	71.9	71.7	64.0
6-bit	NoAdapt	✔	1	44.2	42.0	44.8	39.8	28.9	43.4	34.7	53.2	59.8	59.0	75.1	27.4	39.0	59.1	65.3	47.7
	ZOA	✗	2	47.5	45.5	46.8	41.2	37.5	47.9	41.9	55.6	60.7	60.4	75.6	14.8	49.0	62.3	66.5	50.2
	FOZO	✗	2	41.9	43.4	44.6	41.1	34.8	46.1	40.3	56.6	61.2	64.0	75.1	26.6	47.2	59.7	67.0	50.0
	FOA	✗	28	53.2	51.8	54.6	49.6	38.8	51.0	44.8	60.3	65.0	68.8	76.7	39.5	46.6	67.3	68.6	55.8
	FOA†	✔	28	47.4	49.2	49.6	47.1	32.4	47.5	38.1	40.4	60.7	36.0	74.8	16.4	45.9	61.1	65.6	47.5
	EVA-0 (ours)	✗	2	54.1	55.4	56.0	51.0	48.8	55.9	53.7	63.7	65.4	69.3	78.1	45.6	61.3	69.0	70.8	**59.9**
	EVA-0† (ours)	✔	2	52.0	53.1	54.2	48.8	47.5	55.2	51.4	61.4	63.4	67.1	77.3	43.0	58.5	67.6	68.8	57.9

which requires neither BP nor source statistics. On ViT-Base, EVA-0† achieves 64.6%, outperforming FOA† (55.9%) and even surpassing the BP-based SAR baseline (62.7%). These confirm the benefit of our informative direction estimation (Sec. 4.1) and shortcut-resistant objective design (Sec. 4.2).

Long-term continual test-time optimization A practical TTO method should remain effective over online streams to continuously learn from evolving test-time experience. We further conduct experiments under the continual TTA setting, where the model evolves across domains without reset, which stresses long-term TTO stability. As shown in Table 2, most existing BP-based and ZO-based methods significantly degrade in this setting, *i.e.*, DeYO degrades from 65.7% to 58.5%, and FOA† degrades from 55.9% to 22.3% w.r.t. accuracy on ViT-Base. In contrast, EVA-0 maintains and further improves performance, achieving the best average accuracy across models. Notably, EVA-0 and EVA-0† achieve 45.6% and 41.6% average accuracy on ResNet50-GN, giving the best and second-best results. This stability mainly benefits from anchor-guided optimization, as in Table 4, which allows EVA-0 to accumulate useful test-time signals without being dominated by noisy updates.

Results on quantized models For practical edge deployment, deep models are often quantized before inference, which renders BP infeasible for TTO. We further evaluate the effectiveness of EVA-0 on quantized models in Table 3. EVA-0 significantly outperforms prior forward-only methods on both

Table 4: Ablation study on ResNet50-GN. E^{SR} and ℓ_{SWA} are from Eqn. (8). AGO is anchor-guided optimization, and SSD is sample-wise direction estimation. Single /Continual denotes settings in Tables 1/2.

E^{SR}	ℓ_{SWA}	*AGO*	*SSD*	Single	Continual
NoAdapt				31.4	31.4
ZOA				37.3	36.5
✔				39.5	29.2
	✔			37.5	32.8
✔	✔			41.3	35.4
✔	✔	✔		41.1	42.2
✔			✔	41.1	30.8
	✔		✔	42.6	40.1
✔	✔		✔	44.9	40.8
✔	✔	✔	✔	45.4	45.6

Table 5: Comparison of **throughput (FPS)** and **peak memory** on an NVIDIA Jetson Orin Nano with ViT-Base. Acc. (%) is averaged under single-domain TTA on ImageNet-C (level 5). #FP/#BP: forward/backward passes per sample. OOM denotes out of memory.

Method	BP	FTTA	#FP	#BP	Acc.	FPS	Mem. (MB)
NoAdapt	✗	✔	1	0	55.5	28.96	917
TENT	✔	✔	1	1	59.6	14.03	3,425
CoTTA	✔	✔	3or35	1	61.7	OOM	OOM
SAR	✔	✔	[1, 2]	[0, 2]	62.7	7.02	3,425
DeYO	✔	✔	[1, 2]	[0, 1]	65.7	10.05	3,186
ZOA	✗	✗	2	0	61.3	14.70	918
FOZO	✗	✗	2	0	61.7	14.58	923
FOA	✗	✗	28	0	66.3	1.00	924
EVA-0 (ours)	✗	✗	2	0	66.7	14.08	919
EVA-0†(ours)	✗	✔	2	0	64.6	14.08	919

Table 6: Efficacy on **black-box** model. EVA-0$^{\clubsuit}$ indicates a variant that uses only Eqn. (6) to update input prompts. We report Acc.(%) over ImageNet-C (level 5) with ViT-B per Table 1.

Method	#FP=2	#FP=8	#FP=14	#FP=20	#FP=26
FOA†	58.8	57.6	55.9	55.0	55.9
EVA-0$^{\clubsuit}$ (ours)	64.1	65.2	65.7	65.8	66.1

Table 7: Analysis on anchor design for EVA-0. The anchor is updated: $\theta^{\mathrm{anc}} \leftarrow m\theta_t + (1-m)\theta^{\mathrm{anc}}$. $m = 0$ corresponds to a fixed anchor. Single and Continual denote settings used in Tables 1 & 2.

	m=1e-1	1e-2	1e-3	1e-4	1e-5	0
Single	45.6	45.5	45.5	45.4	45.4	45.4
Continual	40.2	40.5	43.0	45.5	45.6	45.6

8-bit and 6-bit ViT, whereas EVA-0 enables a 6-bit ViT to outperform a non-adapted 32-bit ViT by 4.4% in OOD accuracy. This suggests that EVA-0 offers a novel *deployment tradeoff*: instead of relying on a fixed full-precision model for generalization, a quantized model can recover and even surpass full-precision robustness through lightweight forward-only evolution at testing. Notably, even without using source feature statistics in a fully TTA manner, EVA-0† outperforms FOA by 0.5% on 8-bit ViT and 2.1% on 6-bit ViT, with a *14*-fold forward evaluation reduction. These results collectively underscore the superiority of our EVA-0 in such quantized model deployment scenarios.

5.2 More Discussions

Effectiveness of components in EVA-0 We ablate the key components of EVA-0 on ResNet50-GN in Table 4. First, E^{SR} improves single-domain accuracy from 37.3% (ZOA) to 39.5%, verifying that it provides stronger feedback for ZO updates. Second, ℓ_{SWA} provides a complementary signal, and combining it with E^{SR} improves single-domain accuracy to 41.3%. Third, sample-wise direction estimation (SSD) better exploits the two-forward budget, improving $E^{\mathrm{SR}}+\ell_{\mathrm{SWA}}$ from 41.3% to 44.9% in the single-domain setting. Finally, anchor-guided optimization (AGO) is crucial for continual evolution: adding AGO improves the continual accuracy from 40.8% to 45.6% by controlling the drift accumulated from random ZO updates, without which E^{SR} alone even falls below NoAdapt (29.2% versus 31.4%). With all components, EVA-0 achieves the best results in both settings.

Computation and memory efficiency As shown in Table 5, EVA-0 achieves the best accuracy-efficiency trade-off on the edge device. With only two forward passes and no backward pass, it obtains the highest accuracy of 66.7% at 14.08 FPS with 919MB memory. Compared with BP-based DeYO, EVA-0 improves accuracy by 1.0 point, raises throughput from 10.05 to 14.08 FPS, and lowers memory from 3,186MB to 919MB, while CoTTA even runs out of memory. Compared with ZOA/FOZO, EVA-0 keeps nearly the same throughput and memory but improves accuracy by about 5%. Compared with FOA, EVA-0 achieves higher accuracy with $14\times$ fewer forward passes and $14\times$ higher throughput. Results on an A100 GPU show similar trends (Table E in Appendix E). Overall, EVA-0 makes two-forward TTO practical with strong accuracy and low memory and compute cost.

Efficacy of EVA-0 on black-box models We further evaluate EVA-0 in a black-box setting, where model weights are inaccessible and only input prompts are updated. As shown in Table 6, FOA† does not benefit from additional forward evaluations, degrading from 58.8% at #FP=2 to around 55% with larger budgets. In contrast, EVA-0$^{\clubsuit}$, using only the shortcut-resistant entropy in Eq. (6), achieves

Table 8: **Generality** of our sample-wise perturbation and objective for other ZO methods. We apply them to CAZO [55], while keeping its learnable parameters and curvature-aware sampling on ImageNet-C & ViT-B.

Method	Sample-wise Perturbation	Our Objective	#FP	Acc.
CAZO (K=1)	✗	✗	3	61.9
CAZO (K=2)	✗	✗	5	65.2
CAZO (K=1) + ours	✔	✔	3	**67.3**

Table 9: Sensitivity to **batch size** B. We report the average accuracy (%) on ImageNet-C (level 5) with ResNet50-GN under single-domain TTA.

Method	Batch size B			
	1	8	64	256
ZOA	0.1	37.9	37.3	36.2
FOZO	0.1	35.6	33.8	25.1
EVA-0 (ours)	**45.4**	**45.5**	**45.4**	**45.2**

64.1% with two forward passes and steadily improves to 66.1% at #FP=26. This yields a 5.3% gain under the strict two-forward constraint and a 10.2% gain at #FP=26 over FOA†, showing that our objective provides stable source-free feedback that enables effective black-box test-time evolution.

Analysis on anchor design Table 7 studies the anchor update rate m. In the single-domain setting, EVA-0 is robust to this choice, with all variants achieving around 45.4–45.6% accuracy, where a moving anchor slightly improves the performance. In continual TTO, however, a fast-moving anchor significantly hurts stability, *i.e.*, m=10^{-1} and 10^{-2} only obtain 40.2% and 40.5%, respectively. This indicates that if the anchor follows the online model too quickly, it absorbs noisy updates and loses its role as a stable reference. In contrast, a slowly moving or fixed anchor achieves the best continual accuracy, with $m = 10^{-5}$ and $m = 0$ both reaching 45.6%. Thus, a persistent anchor is important to avoid accumulated random drift. For simplicity, we use a fixed pretrained anchor in our EVA-0.

Generality of our designs to other ZO methods Our sample-wise perturbation and shortcut-resistant objective are not tied to the specific update rule of EVA-0, but are *general and broadly applicable*. To verify this, we plug them into CAZO [55], a recent ZO-TTA method that samples curvature-aware anisotropic perturbations with K symmetric directions, while keeping its learnable parameters and perturbation sampling unchanged. As in Table 8, with the same 3 forward passes (K=1, including inference), our designs significantly boost CAZO from 61.9% to 67.3%, even surpassing CAZO with K=2 (65.2%) that requires 5 forward passes. This indicates that directional exploration and objective shortcuts are common bottlenecks of existing ZO-TTA methods rather than issues specific to our framework. Consequently, our designs are orthogonal to how perturbations are sampled and can be seamlessly plugged into other ZO-based TTA methods for better accuracy and query efficiency.

Robustness to batch size We show that EVA-0 is insensitive to the batch size B. To keep the anchor-guided frequency consistent across batch sizes, our default k=1 is defined for B=64, *i.e.*, one anchor-guided perturbation per 64 samples, and for other B we maintain a sample counter to apply it every 64 samples. As shown in Table 9, ZOA and FOZO collapse at B=1 (0.1%), as their batch-level feature alignment becomes ill-defined with a single sample, and they also degrade as the batch grows, *e.g.*, FOZO drops from 35.6% at B=8 to 25.1% at B=256, since a shared perturbation explores only one direction per batch. In contrast, EVA-0 remains stable at 45.2–45.5% from B=1 to B=256. This is because sample-wise perturbations explore one direction per sample, thus the same samples yield equal direction explorations under any B, while EMA-based sample-wise feature alignment remains well-defined for a single sample. Thus, EVA-0 suits both streaming and large-batch deployment.

6 Conclusion

In this paper, we establish that effective test-time model evolution is possible under an extremely restrictive yet deployment-realistic regime: only two forward passes per sample, without any back-propagation. We reveal that zeroth-order test-time optimization is fundamentally challenged by shortcut exploitation, accumulated weight drift, and less effective update direction estimation, and we address these challenges through EVA-0. By unifying shortcut-resistant objectives, anchor-guided trajectory control, and sample-wise symmetric two-sided perturbations, EVA-0 converts the minimal two-forward budget into both reliable inference and stable online learning. Across full-precision models, quantized networks, continual streams, and black-box settings, EVA-0 consistently delivers strong accuracy, low memory cost, and robust long-term stability. These findings demonstrate that forward-only model evolution is not merely a compromise for restricted scenarios, but a promising and scalable path toward adaptive deployed intelligence.

Acknowledgments

This work is supported by a Start-up Grant from Nanyang Technological University, the Joint WeBank-NTU Research Institute on Fintech, Nanyang Technological University, Singapore, Jinan-NTU Green Technology Research Institute (GreenTRI), and the Centre of AI in Medicine (C-AIM), Nanyang Technological University, Singapore.

References

[1] A. Bartler, A. Bühler, F. Wiewel, M. Döbler, and B. Yang. Mt3: Meta test-time training for self-supervised test-time adaption. In *International Conference on Artificial Intelligence and Statistics*, pages 3080–3090. PMLR, 2022.

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

[3] H. Cai, D. McKenzie, W. Yin, and Z. Zhang. Zeroth-order regularized optimization (zoro): Approximately sparse gradients and adaptive sampling. *SIAM Journal on Optimization*, 32(2):687–714, 2022.

[4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[5] A. Chen, Y. Zhang, J. Jia, J. Diffenderfer, K. Parasyris, J. Liu, Y. Zhang, Z. Zhang, B. Kailkhura, and S. Liu. Deepzero: Scaling up zeroth-order optimization for deep model training. In *International Conference on Learning Representations*, 2024.

[6] S. Cheng, G. Wu, and J. Zhu. On the convergence of prior-guided zeroth-order optimization algorithms. In *Advances in Neural Information Processing Systems*, volume 34, pages 14620–14631, 2021.

[7] W. Choi, D.-Y. Kim, J. Park, J. Lee, Y. Park, D.-J. Han, and J. Moon. Adaptive energy alignment for accelerating test-time adaptation. In *The Thirteenth International Conference on Learning Representations*, 2024.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[9] Z. Deng, G. Chen, S. Niu, H. Luo, S. Zhang, Y. Yang, R. Chen, W. Luo, and M. Tan. Test-time model adaptation for quantized neural networks. *arXiv preprint arXiv:2508.02180*, 2025.

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

[11] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

[12] E. Dupoux, Y. LeCun, and J. Malik. Why ai systems don't learn and what to do about it: Lessons on autonomous learning from cognitive science. *arXiv preprint arXiv:2603.15381*, 2026.

[13] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[14] A. Flaxman, A. Kalai, and H. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394, 2005.

[15] Y. Gandelsman, Y. Sun, X. Chen, and A. Efros. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*, volume 35, pages 29374–29385, 2022.

[16] J. Gao, J. Zhang, X. Liu, T. Darrell, E. Shelhamer, and D. Wang. Back to the source: Diffusion-driven adaptation to test-time corruption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11786–11796, 2023.

[17] T. Gautam, Y. Park, H. Zhou, P. Raman, and W. Ha. Variance-reduced zeroth-order methods for fine-tuning language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 15180–15208, 2024.

[18] S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

[19] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, pages 1–14, 2018.

[20] S. Goyal, M. Sun, A. Raghunathan, and J. Z. Kolter. Test time adaptation via conjugate pseudo-labels. *Advances in Neural Information Processing Systems*, 35:6204–6218, 2022.

[21] W. Gu, L. Gu, Z. Wang, C. Y. Suen, and Y. Wang. Docttt: Test-time training for handwritten document recognition using meta-auxiliary learning. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 1904–1913, 2025.

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

[23] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[25] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, pages 1–11, 2019.

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

[27] W. Jeong, J. Cho, Y. Yoon, and K.-J. Yoon. Synchronizing task behavior: Aligning multiple tasks during test-time training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24340–24350, 2025.

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

[29] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664, 2021.

[30] J. Lee, D. Jung, S. Lee, J. Park, J. Shin, U. Hwang, and S. Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *International Conference on Learning Representations*, 2024.

[31] J.-H. Lee. Parameter dynamics of online machine learning and test-time adaptation. In *Advances in Neural Information Processing Systems*, 2025.

[32] J.-H. Lee and J.-H. Chang. Continual momentum filtering on parameter space for online test-time adaptation. In *International Conference on Learning Representations*, 2024.

[33] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[34] J. Liang, R. He, and T. Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, pages 1–34, 2024.

[35] J. Liang, D. Hu, and J. Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039, 2020.

[36] J. Liu, R. Xu, S. Yang, R. Zhang, Q. Zhang, Z. Chen, Y. Guo, and S. Zhang. Continual-mae: Adaptive distribution masked autoencoders for continual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28653–28663, 2024.

[37] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

[38] S. Malladi, T. Gao, E. Nichani, A. Damian, J. D. Lee, D. Chen, and S. Arora. Fine-tuning language models with just forward passes. In *Advances in Neural Information Processing Systems*, pages 53038–53075, 2023.

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

[40] S. Niu, J. Wu, Y. Zhang, Z. Wen, Y. Chen, P. Zhao, and M. Tan. Towards stable test-time adaptation in dynamic wild world. In *Internetional Conference on Learning Representations*, pages 1–14, 2023.

[41] D. Osowiechi, G. A. V. Hakim, M. Noori, M. Cheraghalikhani, A. Bahri, M. Yazdanpanah, I. Ben Ayed, and C. Desrosiers. Nc-ttt: A noise constrastive approach for test-time training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6078–6086, 2024.

[42] D. Osowiechi, G. A. V. Hakim, M. Noori, M. Cheraghalikhani, I. Ben Ayed, and C. Desrosiers. Tttflow: Unsupervised test-time training with normalizing flow. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2126–2134, 2023.

[43] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision*, 2019.

[44] D. Silver and R. S. Sutton. Welcome to the era of experience. *Google AI*, 1:11, 2025.

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

[46] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248, 2020.

[47] J.-A. Termöhlen, M. Klingner, L. J. Brettin, N. M. Schmidt, and T. Fingscheidt. Continual unsupervised domain adaptation for semantic segmentation by online frequency domain style transfer. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2881–2888. IEEE, 2021.

[48] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, pages 1–12, 2021.

[49] Q. Wang, O. Fink, L. Van Gool, and D. Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

[50] X. Wang and T. Wang. Fozo: Forward-only zeroth-order prompt optimization for test-time adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2026.

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

[52] F. You, J. Li, and Z. Zhao. Test-time batch statistics calibration for covariate shift. *arXiv preprint arXiv:2110.04065*, 2021.

[53] Y. Yuan, B. Xu, L. Hou, F. Sun, H. Shen, and X. Cheng. Tea: Test-time energy adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 23901–23911, 2024.

[54] Z. Yuan, C. Xue, Y. Chen, Q. Wu, and G. Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *European Conference on Computer Vision*, pages 191–207. Springer, 2022.

[55] J. Zhang, S. Yin, P. Liu, R. Ying, and F. Wen. Curvature-aware zeroth-order optimization for memory-efficient test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 836–846, 2026.

[56] M. M. Zhang, S. Levine, and C. Finn. Memo: Test time robustness via adaptation and augmentation. In *Advances in Neural Information Processing Systems*, pages 38629–38642, 2022.

Supplementary Materials for " EVA-0: Test-Time Model Evolution with Only Two Forward Passes per Sample "

Contents

A Pseudo-Code of EVA-0 **16**

B Related Work **17**

C More Design Details of EVA-0 **18**

C.1 Computational- and Memory-Efficient Implementation of EVA-0 18

C.2 Balancing Inward and Outward Updates with Anchor-Guided Perturbation 19

D More Implementation Details **19**

D.1 More Details on Datasets 19

D.2 More Experimental Protocols on Evaluation 19

E Additional Discussions **21**

E.1 Limitations, Future Directions, and Broader Impacts 25

A Pseudo-Code of EVA-0

In this appendix, we provide the pseudo-code of our EVA-0 method. As shown in Algorithm 1, for each online test mini-batch $\mathcal{B}_t$, we first compute the anchor direction d_t from the current online model to the anchor model, and sample perturbations from an adaptive distribution $\mathcal{D}_t$ (refer to lines 3–6). For each test sample x_i, EVA-0 performs two symmetric forward passes with the positively and negatively perturbed models, $f_{\theta_t+\mu z_{t,i}}$ and $f_{\theta_t-\mu z_{t,i}}$ (refer to line 7). These two forward passes are used for both inference and adaptation: the prediction is obtained by averaging the two perturbed outputs (refer to line 8), while the update direction is estimated from their shortcut-resistant loss difference (refer to lines 9-10). Specifically, the shortcut-resistant objective ℓ_{sr} suppresses degenerate entropy-reduction shortcuts, and the sample-wise two-sided difference provides an informative zeroth-order direction under the two-forward constraint. After aggregating the sample-wise directions over the mini-batch, we update the online model with a zeroth-order gradient step (refer to lines 11-12). Lastly, we apply a weak weight relaxation toward the anchor model to softly bound the online trajectory and mitigate accumulated weight drift (refer to line 13). The online states, including the output center and target moments, are updated by EMA for subsequent test batches (refer to line 14).

Algorithm 1: EVA-0: Two-Forward Test-Time Model Evolution

Input: Test stream $\{\mathcal{B}_t\}_{t=1}^{T}$, model f_{θ_0}, learning rate η, hyper-parameters $\gamma = 0.999$, $k = 1$, μ.
Output: Predictions $\{\hat{y}_{t,i}\}$

1 Initialize $\theta_1 \leftarrow \theta_0$, $\theta^{\mathrm{anc}} \leftarrow \theta_0$, and online states $c_1, m_1, q_1 = \emptyset$
2 **for** $t = 1, \ldots, T$ **do**
3 Receive $\mathcal{B}_t = \{x_i\}_{i=1}^{B}$ and compute $d_t = \theta^{\mathrm{anc}} - \theta_t$
4 Select anchor-guided subset $\mathcal{I}_t$ with $|\mathcal{I}_t| = k$
5 **for** $i = 1, \ldots, B$ **do**
6 Sample $z_{t,i} \sim \mathcal{D}_t$ using random or anchor-guided perturbation `// Eqn. (10)`
7 $(o^+_{t,i}, h^+_{t,i}) = f_{\theta_t+\mu z_{t,i}}(x_i)$, $(o^-_{t,i}, h^-_{t,i}) = f_{\theta_t-\mu z_{t,i}}(x_i)$ `// two forwards`
8 $\hat{y}_{t,i} = \frac{1}{2}(o^+_{t,i} + o^-_{t,i})$ `// inference`
9 $\ell^+_{t,i} = \ell_{\mathrm{sr}}(o^+_{t,i}, h^+_{t,i})$, $\ell^-_{t,i} = \ell_{\mathrm{sr}}(o^-_{t,i}, h^-_{t,i})$ `// shortcut-resistant loss, Eqn. (8)`
10 $g_{t,i} = \frac{\ell^+_{t,i} - \ell^-_{t,i}}{2\mu} z_{t,i}$ `// two-sided ZO direction`
11 $\widehat{g}^{\mathrm{EVA}}_t = \frac{1}{B}\sum_{i=1}^{B} g_{t,i}$
12 $\theta'_{t+1} = \theta_t - \eta \widehat{g}^{\mathrm{EVA}}_t$
13 $\theta_{t+1} = (1-\gamma)\theta'_{t+1} + \gamma\theta^{\mathrm{anc}}$ `// weight relaxation, Eqn. (11)`
14 Update $c_{t+1}, m_{t+1}, q_{t+1}$ by EMA via $\bar{o}_t = (o^+_t + o^-_t)/2$, $\mathbb{E}[h = (h^+_t + h^-_t)/2]$, $\mathbb{E}[h^2 = (h^{+^2}_t + h^{-^2}_t)/2]$ using moving average factor of 0.9

B Related Work

Backpropagation-based TTA TTT [46] first pioneers BP-based TTA. During the training phase, TTT jointly trains the network with an auxiliary self-supervised branch, instantiated as rotation prediction [19], and updates the shared encoder via the self-supervised objective at test time. The following works [15, 36, 41, 1] extend TTT from two axis: 1) Auxiliary task design: TTT-MAE [15] and CTMAE [36] replace rotation prediction with masked image reconstruction [23] and histogram-of-oriented-gradients reconstruction. NC-TTT [41] and TTTFlow [42] introduce a discriminator on source and OOD features to pull features back to the source at testing. 2) Boosting correlations between the auxiliary and the main task: MT3 [1] and Doc-TTT [21] exploit a meta-learning approach (MAML [13]) to improve the gradient alignment between the auxiliary and main tasks. S4T [27] further synchronizes multiple downstream tasks during TTT by predicting inter-task relations.

TTT requires altering the model training process and access to the source data, which hinders its applicability to privacy-sensitive scenarios and off-the-shelf models. To address this, Fully TTA methods update any given model via unsupervised learning objectives, such as entropy minimization [48, 30, 40], prediction consistency maximization [56, 49], energy adaptation [53, 7], pseudo labeling [35, 20], *etc.* However, these TTA methods still rely on backpropagation, which introduces substantial memory overhead, and can be incompatible with resource-limited edge devices, quantized models, specialized accelerators, or black-box model access settings.

Forward-only & optimization-free TTA Without using backpropagation and optimizing model parameters, these methods seek model adaptation alone without any online learning at all. Existing methods can be categorized into three paradigms: 1) Statistics calibration [52], which replaces the source training statistics with statistics computed from the test data. 2) Input purification, which denoises the test data back to the source distribution via diffusion models [16] or Fast Fourier Transform [47]. 3) Direct output adjustment, which maintains online class prototypes to replace the original classifier head for prediction [26], or calibrates output according to estimated priors [2]. However, without updating parameters online, these methods suffer from limited learning capability and fail to effectively transform unlabeled test-time experiences into adaptive knowledge, often resulting in limited performance gain when handling out-of-distribution data.

Forward-optimization TTA replaces the gradient-based parameter updates with derivative-free optimization strategies that require only forward evaluations. Recently, FOA explores this direction using CMA-ES [22] to evolve input prompts, which online maintains the covariance of update directions to guide future sampling. However, this process requires many forward evaluations, *e.g.*, $K = 28$ for stable learning signals, and is limited to a low-dimensional parameter space due to the high memory cost of the covariance matrix. ZOA [9] instead adopts one-sided SPSA [45] to adapt quantized models using only two forward passes per sample. However, it leaves a significant performance gap compared to FOA, as shown in Table 1, due to its coarse and noisy signal from one-sided SPSA. To this end, we study *test-time model evolution* under a strict two-forward constraint. We reveal the limitations of prior ZO methods from the objective design, direction estimation, and weight dynamics, and propose EVA-0 to substantially advance the effectiveness and stability of two-forward model evolution.

Zeroth-order optimization (ZOO) estimates update directions from function-value queries, typically via finite differences [37], and thus avoids backpropagation. This makes it memory-efficient and suitable for memory-resource-constrained, quantized, or black-box models. Prior works have established its theoretical foundations [14, 18, 11]. However, ZOO often suffers from high-variance gradient estimates and slow convergence in deep, high-dimensional models. To improve its efficiency, existing studies have made several attempts, such as sampling [6] and sparsity-aware updates [3], with recent efforts further scaling ZOO to large models [38].

However, existing ZOO methods [38, 5] are still mainly designed for offline supervised settings, where many forward evaluations are affordable. In contrast, we study ZOO in a stricter test-time deployment setting, where only two forward passes are available for each test sample to support both inference and online adaptation. This setting is highly challenging yet under-explored due to its stringent efficiency constraint and unsupervised nature. We analyze the fundamental limitations of ZOO in this two-forward test-time model evolution setting and derive key insights to guide the design of more effective and stable methods.

C More Design Details of EVA-0

C.1 Computational- and Memory-Efficient Implementation of EVA-0

EVA-0 uses sample-wise independent perturbations, which could be naively implemented by constructing a different perturbed model for each sample. This implementation is inefficient: it would either serialize the forward computation over samples, or materialize a large perturbation tensor of size $B \times |\theta|$. We avoid this overhead with a layer-wise seeded implementation that preserves batch parallelism for most computations and only applies sample-wise perturbations to the learnable layers.

Parallel computation Let $\theta = (\phi, \psi)$, where ϕ denotes frozen parameters and ψ denotes the adapted parameters. EVA-0 preserves standard batched inference for the frozen part:

$$X = f_\phi(\mathcal{B}). \tag{12}$$

It then applies sample-wise perturbations only to the adapted part:

$$y_i^{\pm} = f_{\phi,\, \psi \pm \mu z_i}(x_i), \qquad i = 1, \ldots, B. \tag{13}$$

In implementation, this does not require running B independent models. Since ψ contains only a small subset of layers, such as LayerNorm parameters, we form batched perturbed weights for these layers and compute them in parallel. Thus, EVA-0 keeps the efficiency of standard mini-batch inference for frozen layers, while introducing sample-wise perturbations only on a small fraction of layers where test-time optimization is performed, which introduces a minimal computation cost.

Seeded layer-wise perturbation A naive implementation of sample-wise SPSA would store the full perturbation matrix $Z \in \mathbb{R}^{B \times |\theta|}$ for the mini-batch. This can dominate memory when the number of optimized parameters grows. Inspired by MeZO [38], EVA-0 instead stores only random seeds. For each adapted layer ℓ, we deterministically regenerate $z_{\ell,i}$ from the global seed and the layer index, use it to compute the perturbed layer output, and immediately discard it.

Formally, let s_t be the random seed at step t. For each adapted layer ℓ, EVA-0 samples

$$Z_{\ell,t} = \mathrm{Rand}(s_t, \ell) \in \mathbb{R}^{B \times |\theta_\ell|},$$

uses $Z_{\ell,t}$ to compute the two perturbed forward passes, and then releases it before moving to the next layer. During gradient construction, EVA-0 regenerates the same $Z_{\ell,t}$ using the stored seed and multiplies it by the corresponding finite-difference scalar feedback.

Memory cost analysis With the above implementation, the additional memory of EVA-0 is proportional to the largest adapted layer, rather than the total number of adapted parameters:

$$O\left(B \max_\ell |\theta_\ell|\right) \quad \text{instead of} \quad O(B|\theta|).$$

Moreover, since EVA-0 is forward-only, it does not store intermediate activations for backpropagation. As a result, EVA-0 keeps the memory footprint close to standard inference while still supporting sample-wise independent perturbations and two-sided SPSA updates. In practice, after each adapted layer computes its perturbed output, the sampled perturbation and temporary batched parameters are *immediately deleted*, and the perturbation is later regenerated from the seed to compute the gradient.

Table A: Comparison of wall-clock time and memory for processing 50,000 ImageNet-C images on an A100 GPU with ViT-Base. Both variants lead to the same performance.

	Time (s)	Memory (MB)
Memory-efficiency-oriented implementation	241	822
Computation-efficiency-oriented implementation	236	828

Table 5 in the main paper follows the implementation as proposed in Sec. C.1. The main extra computation cost, beyond doubling the forward passes, comes from the resampling of perturbations for three times: $f_{\theta_t + \mu z_{t,i}}(x_i)$, $f_{\theta_t - \mu z_{t,i}}(x_i)$, and gradient computation. One can trade memory efficiency for computational efficiency by sampling perturbation once, storing the perturbation, and reusing it for each sample. As shown in Table A, this leads to a reduction of 5 seconds in latency. In practice, one can flexibly choose the appropriate implementation according to the used scenarios.

C.2 Balancing Inward and Outward Updates with Anchor-Guided Perturbation

Although anchor-guided perturbations provide recovery-aware directions, repeated ZO updates may still accumulate biased drift away from the anchor. Since outward movement is riskier in unsupervised online evolution, EVA-0 further balances updates that move toward and away from the anchor.

Let $\Delta_t = -\eta \hat{g}_t$ be the ZO update and $e_t = (\theta^{\text{anc}} - \theta_t)/(\|\theta^{\text{anc}} - \theta_t\|_2 + \epsilon)$ be the unit direction toward the anchor. We use $c_t = \langle \Delta_t, e_t \rangle$ to measure whether the update moves inward ($c_t > 0$) or outward ($c_t < 0$). EVA-0 maintains EMA estimates of recent inward and outward magnitudes:

$$a_t^{\text{in}} = \beta a_{t-1}^{\text{in}} + (1-\beta)\max(c_t, 0), \qquad a_t^{\text{out}} = \beta a_{t-1}^{\text{out}} + (1-\beta)\max(-c_t, 0). \tag{14}$$

When the current update moves away from the anchor, EVA-0 rescales only its anchor-parallel outward component by

$$\alpha_t = \min\left(1, \frac{a_t^{\text{in}}}{a_t^{\text{out}} + \epsilon}\right), \qquad \widetilde{\Delta}_t = \Delta_t + (\alpha_t - 1) c_t e_t \quad \text{if } c_t < 0. \tag{15}$$

For inward updates, we simply set $\widetilde{\Delta}_t = \Delta_t$.

This strategy does not prevent the model from moving away from the anchor. It only weakens outward movement when recent outward updates dominate inward corrections to ensure long-term stability.

D More Implementation Details

D.1 More Details on Datasets

We evaluate the test-time OOD generalization ability of all methods on a large-scale and widely used benchmark, namely **ImageNet-C**[2] [25]. ImageNet-C is constructed by corrupting the original ImageNet [8] test set. The corruption (as shown in Figure A) consists of 15 different types, *i.e.*, Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transformation, pixelation, and JPEG compression, where each corruption type has 5 severity levels and the larger severity level means more severe distribution shift.

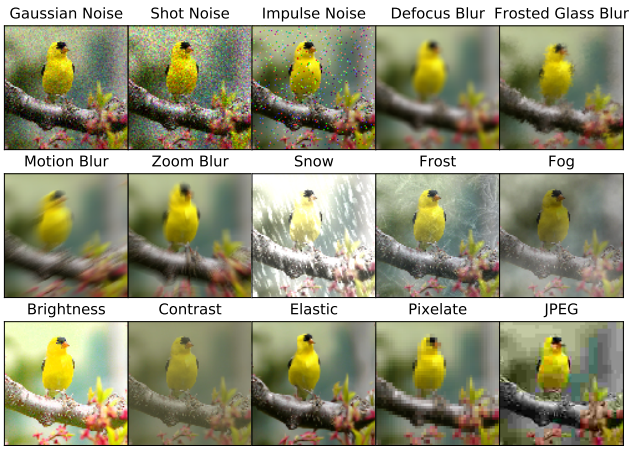


Figure A: Visualizations of different corruption types in ImageNet-C benchmark.

D.2 More Experimental Protocols on Evaluation

All pre-trained models involved in our paper are publicly available, including ResNet50-GN[3] and ViT-Base[4] obtained from `timm` repository [51]. In the following, we provide the implementation

[2] `https://zenodo.org/record/2235448#.YzQpq-xBxcA`

[3] `https://github.com/rwightman/pytorch-image-models/releases/download/v0.1-rsb-weights/resnet50_gn_a1h2-8fe6c4d0.pth`

[4] `https://storage.googleapis.com/vit_models/augreg/B_16-i21k-300ep-lr_0.001-aug_medium1-wd_0.1-do_0.0-sd_0.0--imagenet2012-steps_20k-lr_0.01-res_224.npz`

details of our proposed method and all comparative methods evaluated in our experiments, which help reproduce our results.

EVA-0 (Ours). We use source pretrained parameters as θ^{anc} and set μ to 0.06 in Eqn. (5), k to 1 in Eqn. (10), γ to 0.001 in Eqn. (11) and ρ to 0.999 in Eqn. (7), consistently for all experiments. We use SGD as the update rule, with a learning rate of 0.002 and a batch size of 64 for both models. For ViT-Base, we configure λ to 500, and update online statistics c_t, m_t, q_t using EMA with a factor of 0.9. For ResNet50-GN, we configure λ to 50, and update online statistics c_t, m_t, q_t using EMA with a factor of 0. Following ZOA, the trainable parameters are the affine parameters of norm layers from layer 1 to layer 3 in ResNet50-GN, from blocks 2 to blocks 8 in ViT-Base.

TENT[5] [48]. We follow all hyper-parameters that are set in TENT unless it does not provide. We use SGD as the update rule, with a momentum of 0.9, batch size of 64, and learning rate of 0.00025 / 0.001 for ResNet50-GN / ViT-Base. The trainable parameters are all the affine parameters of normalization layers for both models.

SAR[6] [40]. We follow all hyper-parameters that are set in SAR unless it does not provide. We use SGD as the update rule, with a momentum of 0.9, batch size of 64, and learning rate of 0.00025 / 0.001 for ResNet50-GN / ViT-Base. The entropy threshold E_0 is set to $0.4\times \ln C$, where C is the number of task classes. The trainable parameters are the affine parameters of norm layers from layer 1 to layer 3 in ResNet50-GN, from blocks 1 to blocks 8 in ViT-Base.

DeYO[7] [30]. We follow all hyper-parameters that are set in DeYO unless it does not provide. Specifically, the entropy constant E_0 (for reliable sample identification) is set to $0.4 \times \ln 1000$, and the factor τ_{Ent} is set to $0.5\times \ln 1000$. The Pseudo-Label Probability Difference (PLPD) threshold τ_{PLPD} is set to 0.2. The update rule is SGD, with a momentum of 0.9, batch size of 64, and learning rate of 0.00025 / 0.001 for ResNet50-GN / ViT-Base. Trainable parameters are the affine parameters of norm layers from layer 1 to 3 in ResNet50-GN, from blocks 1 to 8 in ViT-Base.

FOA[8] [39]. We follow all hyper-parameters that are set in FOA unless it does not provide. On ViT-Base, FOA adapts the model by learning 3 new input prompts, initialized uniformly. The trade-off parameter λ in the fitness function is set to 0.4. On ResNet50-GN, FOA updates an 7×7 convolutions applied before model inputs, initialized with zeros. The trade-off parameter λ in the fitness function is set to $1/64$. Optimization is performed via the derivative-free CMA-ES algorithm with a population size of 28. The trade-off parameter λ in the fitness function is set to 0.4. For activation shifting, the step size γ is 1.0, and historical statistics are updated with an exponential moving average factor $\alpha = 0.1$. The trainable parameters are the prompt embeddings.

ZOA[9] [9]. We follow all hyper-parameters that are set in ZOA unless it does not provide. For the ViT model, perturbation vectors ϵ and ν are added with step sizes 0.02 and 0.05, respectively. The learning rates for parameters θ and α are 0.0005 and 0.01. The trainable parameters include the domain knowledge perturbation vectors, which are the affine parameters of the norm layers from layer 1 to 3 in ResNet50-GN, and from blocks 1 to 8 in ViT-Base.

FOZO[10] [50]. We follow all hyper-parameters that are set in FOZO unless it does not provide. It employs an n-SPSA zero-order gradient estimator, where n is set to 1 for 2 forward passes. As in FOA, on ViT-Base, FOZO adapts the model by learning 3 new input prompts, initialized uniformly. The trade-off parameter λ in the unsupervised loss is 0.4. On ResNet50-GN, updates an 7×7 convolutions applied before model inputs, initialized with zeros. The trade-off parameter λ in the unsupervised loss is $1/64$. The initial learning rate η is 0.08. The decay factor α is 0.9, the threshold factor τ is 1.05, and the moving average factor β for historical average loss is 0.9. Source training statistics are estimated using the validation set of ImageNet-1K. The trainable parameters are the prompt embeddings.

[5] https://github.com/DequanWang/tent
[6] https://github.com/mr-eggplant/SAR
[7] https://github.com/Jhyun17/DeYO
[8] https://github.com/mr-eggplant/FOA
[9] https://github.com/DengZeshuai/ZOA
[10] https://github.com/eVI-group-SCU/FOZO

E Additional Discussions

Ablation on coefficient γ in Eqn. (11). We study the effect of the weak anchor relaxation in Table B. After the ZO update produces an intermediate model θ'_{t+1}, EVA-0 applies Eqn. (11) with relaxation coefficient γ, where a larger γ indicates a stronger pull toward the anchor and $\gamma = 0$ disables this relaxation. When the pull is too strong, *e.g.*, γ=0.1, the model is overly constrained around the anchor and obtains only 39.1% average accuracy. As γ decreases, the model gains more plasticity and the performance improves, reaching the best accuracy of 45.6% at γ=0.001. However, when the relaxation becomes too weak, *e.g.*, γ=0.00001 or γ=0, the average accuracy drops to 43.5%.

Compared with EVA-0 without AGO (37.0%), anchor-guided optimization with the proper relaxation improves continual accuracy by 8.6%. Compared with γ=0, the weak relaxation further brings a 2.1% gain. These results confirm that a small but persistent restoring force is important for long-term TTO, whereas the anchor-guided perturbation also significantly improves the stability, 37.0% $\rightarrow$ 43.5%.

Table B: Ablations of γ on ImageNet-C (severity level 5) regarding **Accuracy (%)** under `continuous adaptation` with R50-GN. AGO is short for anchor-guided optimization. γ=0 disables Eqn. (11).

	Noise			Blur				Weather				Digital				
Method	Gaus.	Shot	Imp.	Def.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elas.	Pix.	JPEG	Avg.
NoAdapt	22.1	23.0	22.0	19.8	11.4	21.5	25.0	40.3	47.0	34.0	68.8	36.2	18.5	29.2	52.6	31.4
EVA-0 (w/o AGO)	35.1	37.5	36.6	21.3	19.9	30.9	38.0	38.8	43.8	48.5	61.4	32.4	33.1	38.8	39.5	37.0
EVA-0 (γ=0.1)	30.4	31.5	30.9	25.1	17.2	29.3	33.2	45.0	46.2	51.9	70.7	42.9	30.0	46.2	55.5	39.1
EVA-0 (γ=0.01)	35.7	37.5	36.6	29.2	20.8	34.9	39.2	49.0	49.7	57.3	72.2	47.7	36.6	51.2	57.2	43.6
EVA-0 (γ=0.001)	37.0	40.4	39.5	28.2	23.1	38.4	45.1	50.3	51.8	60.2	72.3	49.1	40.5	52.6	56.0	45.6
EVA-0 (γ=0.0001)	36.6	39.3	37.9	25.0	24.1	35.4	44.2	49.0	51.6	57.9	71.1	45.8	41.3	50.3	51.5	44.1
EVA-0 (γ=0.00001)	36.5	38.9	37.5	24.4	24.2	35.0	43.7	48.4	51.3	56.9	70.6	44.8	41.2	49.5	50.1	43.5
EVA-0 (γ=0)	36.5	38.9	37.4	24.3	24.2	35.0	43.7	48.4	51.2	56.8	70.5	44.6	41.2	49.3	49.9	43.5

Ablation on the number of anchor-guided perturbations k. We further ablate the number of anchor-guided perturbations k in each mini-batch, as shown in Table C. Without AGO, EVA-0 obtains only 37.0% average accuracy under continual adaptation, showing that fully random ZO updates suffer from long-term drift. Introducing anchor-guided perturbations substantially improves stability: even k=1 improves the average accuracy to 45.6%, bringing an 8.6% gain over the variant without AGO. Increasing k from 1 to 5 yields very similar performance, with accuracy remaining around 45.4%–45.6%. This indicates that only a small amount of anchor-guided exploration is sufficient to provide a stable restoring direction. Therefore, we use k=1 by default, which provides strong stabilization while minimally interfering with random exploration.

Table C: Ablations of k on ImageNet-C (severity level 5) regarding **Accuracy (%)** under `continuous adaptation` with ResNet50-GN. AGO is short for anchor-guided optimization.

	Noise			Blur				Weather				Digital				
Method	Gaus.	Shot	Imp.	Def.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elas.	Pix.	JPEG	Avg.
NoAdapt	22.1	23.0	22.0	19.8	11.4	21.5	25.0	40.3	47.0	34.0	68.8	36.2	18.5	29.2	52.6	31.4
EVA-0 (w/o AGO)	35.1	37.5	36.6	21.3	19.9	30.9	38.0	38.8	43.8	48.5	61.4	32.4	33.1	38.8	39.5	37.0
EVA-0 (k=1)	37.0	40.4	39.5	28.2	23.1	38.4	45.1	50.3	51.8	60.2	72.3	49.1	40.5	52.6	56.0	45.6
EVA-0 (k=2)	36.8	40.3	39.3	27.9	23.3	37.3	44.3	50.3	52.0	59.8	72.3	48.3	41.0	53.6	56.6	45.5
EVA-0 (k=3)	36.6	40.1	39.4	28.4	24.1	37.7	44.5	50.8	51.8	60.4	72.1	48.3	40.4	53.1	55.7	45.6
EVA-0 (k=4)	36.3	40.3	39.3	27.5	22.9	38.0	44.4	49.7	52.1	60.0	72.2	48.4	41.0	53.0	56.4	45.4
EVA-0 (k=5)	36.4	40.2	39.4	27.3	22.9	37.7	44.2	49.7	52.1	60.0	72.3	48.2	41.0	53.3	56.3	45.4

Effectiveness of E^{SR} under imbalanced target distributions. We further evaluate whether the proposed shortcut-resistant entropy remains effective under non-i.i.d. target streams with severe label imbalance. To isolate the effect of the objective, all methods in Table D are optimized with backpropagation. Under the online imbalanced label-shift protocol of SAR [40], conventional entropy minimization becomes highly unstable: Tent drops to 23.8% average accuracy, even worse than NoAdapt (31.4%), and SAR only reaches 33.6%. In contrast, our shortcut-resistant entropy in Eqn. (6) achieves 48.0% average accuracy, improving over Tent, SAR, and DeYO by 24.2%, 14.4%, and 7.6%, respectively. These results show that the benefit of E^{SR} is not limited to ZO optimization: by suppressing shortcut solutions such as confidence amplification and prediction collapse, it also provides a more reliable adaptation signal under challenging non-i.i.d. and imbalanced target distributions.

Table D: Effectiveness of our shortcut-resistant entropy E^{SR} in Eqn. (6) under imbalanced target distributions. Results are reported on ImageNet-C (severity level 5) under **ONLINE IMBALANCED LABEL SHIFTS** (imbalance ratio = ∞) regarding **Accuracy (%)** following SAR [40].

	Noise			Blur				Weather				Digital				
Method	Gaus.	Shot	Imp.	Def.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elas.	Pix.	JPEG	Avg.
NoAdapt	22.1	23.0	22.0	19.8	11.4	21.5	25.0	40.3	47.0	34.0	68.8	36.2	18.5	29.2	52.6	31.4
TENT	6.0	19.0	11.5	13.8	8.6	19.5	17.1	16.4	21.7	1.9	69.8	42.2	6.3	49.6	53.6	23.8
SAR	36.5	38.1	37.0	19.4	18.1	33.4	16.1	22.6	44.4	3.8	71.6	46.6	7.1	52.0	56.5	33.6
DeYO	43.4	44.9	43.3	22.7	24.0	40.5	5.0	52.3	51.0	0.8	73.3	52.4	19.0	58.8	59.0	39.4
E^{SR} (ours)	38.2	41.2	40.0	28.8	28.0	40.2	44.7	56.6	52.0	60.9	72.3	51.2	48.7	59.5	57.7	48.0

Efficiency comparison on an A100 GPU. In addition to the edge-device results in Table 5, we also compare the wall-clock time and memory of different methods on an A100 GPU in Table E, and observe consistent trends. With only two forward passes and no backward pass, EVA-0 obtains the highest accuracy of 66.7% while using 241s runtime and 822MB memory on ViT-Base. Compared with BP-based DeYO, EVA-0 improves accuracy by 1.0 point, reduces runtime from 355s to 241s, and lowers memory usage from 5,350MB to 822MB. Compared with two-forward ZO baselines, EVA-0 keeps nearly the same runtime and memory as ZOA/FOZO, but improves the accuracy by about 5%. Compared with FOA, EVA-0 achieves higher accuracy with $14\times$ fewer forward evaluations and about $15\times$ faster runtime. These results further confirm that EVA-0 combines strong adaptation accuracy with low memory and computation cost across different hardware platforms.

Table E: Comparison of **wall-clock time** and **memory** for processing 50,000 ImageNet-C images on an A100 GPU with ViT-Base. Acc. (%) is averaged under single-domain TTA on ImageNet-C (level 5). #FP and #BP are the number of forward and backward passes per sample.

Method	BP	FTTA	#FP	#BP	Acc.	Time (s)	Mem. (MB)
NoAdapt	✗	✔	1	0	55.5	115	819
TENT	✔	✔	1	1	59.6	245	5,165
CoTTA	✔	✔	3or35	1	61.7	885	16,836
SAR	✔	✔	[1, 2]	[0, 2]	62.7	488	5,166
DeYO	✔	✔	[1, 2]	[0, 1]	65.7	355	5,350
ZOA	✗	✗	2	0	61.3	234	822
FOZO	✗	✗	2	0	61.7	248	830
FOA	✗	✗	28	0	66.3	3597	831
EVA-0 (ours)	✗	✗	2	0	66.7	241	822
EVA-0† (ours)	✗	✔	2	0	64.6	241	822

Comparison with normalization-statistics adaptation. A simple alternative to learning-based TTA is to re-estimate normalization statistics on test data without updating any parameters, *e.g.*, AdaBN [33] and SITA [28]. As shown in Table F, on ResNet-50 (BN), AdaBN and SITA improve NoAdapt from 18.2% to 31.7% and 31.8–32.2%, respectively, while EVA-0 further reaches 34.7%, showing that feedback-driven optimization brings gains beyond statistics re-estimation. Notably, EVA-0 also requires fewer sample-equivalent evaluations than SITA (2 versus 3 or 5). More importantly, statistics adaptation applies only to models with batch normalization, since group and layer normalization do not maintain batch-dependent running statistics. On ResNet-50 (GN) and ViT-Base (LN), where AdaBN and SITA are inapplicable, EVA-0 still improves accuracy from 31.4% to 45.4% and from 55.5% to 66.7%, respectively, demonstrating its broad applicability to modern architectures.

Table F: Comparison with normalization-statistics adaptation methods across architectures. We report average accuracy (%) over 15 corruptions on ImageNet-C (severity level 5). #FP counts sample-equivalent forward evaluations per sample, *e.g.*, SITA with k augmented views costs $k+1$. N/A: not applicable, since GN and LN maintain no batch-dependent running statistics.

Method	#FP	ResNet-50 (BN)	ResNet-50 (GN)	ViT-Base (LN)
NoAdapt	1	18.2	31.4	55.5
AdaBN	1	31.7	N/A	N/A
SITA (k=2)	3	31.8	N/A	N/A
SITA (k=4)	5	32.2	N/A	N/A
EVA-0 (ours)	2	**34.7**	**45.4**	**66.7**

Robustness under online imbalanced label shifts. Following SAR [40], we evaluate EVA-0 under online imbalanced label shifts (imbalance ratio $= \infty$), where the label distribution of the test stream changes abruptly over time. As shown in Table G, TENT collapses on several corruptions (*e.g.*, 1.1% on Fog), and the ZO baselines ZOA, FOZO, and FOA only reach 57.5–62.5%. In contrast, EVA-0 achieves the best average accuracy of 66.0%, outperforming FOA by 3.5% and even the BP-based SAR and DeYO without backpropagation. This benefits from two designs: the decentering term in Eqn. (6) prevents predictions from collapsing to the currently dominant classes, while EMA-based statistics aggregate the whole stream and remain stable under abrupt label changes.

Table G: Comparisons with state-of-the-art methods on ImageNet-C (severity level 5) under **ONLINE IMBALANCED LABEL SHIFTS** (imbalance ratio = ∞) following SAR [40], regarding **Accuracy (%)** with ViT-Base. The **bold** number indicates the best result.

		Noise			Blur				Weather				Digital				
Method	BP	Gaus.	Shot	Imp.	Def.	Glass	Mot.	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	✔	61.3	62.5	62.7	59.7	58.5	64.7	60.9	28.9	67.0	1.1	79.7	67.1	65.4	74.1	71.6	59.0
SAR	✔	59.6	60.7	60.8	57.8	57.6	62.6	59.6	66.4	64.8	68.3	78.7	64.5	65.3	73.1	70.8	64.7
DeYO	✔	58.4	59.5	59.2	54.7	56.5	62.9	61.2	68.2	65.8	68.9	77.8	59.9	69.7	73.7	71.6	64.5
ZOA	✗	56.4	57.1	57.8	50.9	47.5	57.1	51.4	63.8	60.1	64.4	77.4	53.1	56.4	67.5	67.8	59.2
FOZO	✗	55.8	56.2	55.3	50.1	39.6	53.5	47.5	63.6	60.6	64.7	75.9	56.2	50.1	66.7	67.1	57.5
FOA	✗	58.5	59.9	60.2	56.2	50.1	57.7	53.3	66.2	66.7	69.3	77.8	63.1	58.5	69.7	70.9	62.5
EVA-0 (ours)	✗	60.7	62.0	62.6	57.4	55.5	61.7	60.3	69.3	67.8	73.1	79.6	68.0	66.7	72.8	71.8	**66.0**

Robustness under natural domain shifts. Beyond synthetic corruptions, we evaluate EVA-0 on DomainNet [43] with ResNet-50 (GN) under continual TTA, where each domain serves as the source and the model adapts sequentially to the other three domains. As shown in Table H, EVA-0 achieves the best accuracy under every source domain and improves NoAdapt from 59.2% to 62.6% on average, outperforming the BP-free ZOA (60.6%) and even the BP-based TENT, SAR, and DeYO (59.4–61.4%) with only two forward passes. These results show that EVA-0 generalizes from synthetic corruptions to natural domain shifts and remains stable under continual domain changes.

Table H: Comparisons on **DomainNet** with ResNet-50 (GN) under `continuous adaptation` regarding **Accuracy (%)**. Each domain serves as the source, and the model adapts sequentially to the other three domains in the listed order. The **bold** number indicates the best result.

Source:		Real				Painting				Clipart				Sketch				Overall
Method	BP	Clp.	Pnt.	Skt.	Avg.	Rel.	Skt.	Clp.	Avg.	Skt.	Rel.	Pnt.	Avg.	Pnt.	Clp.	Rel.	Avg.	Avg.
NoAdapt	✗	55.0	65.3	47.8	56.0	78.4	51.2	54.3	61.3	52.0	66.2	51.4	56.5	59.3	60.7	68.7	62.9	59.2
TENT	✔	55.1	65.7	49.5	56.8	78.6	51.0	55.0	61.6	52.1	66.3	52.1	56.9	59.0	60.4	67.3	62.2	59.4
SAR	✔	55.1	65.7	49.5	56.8	78.5	51.5	55.0	61.7	52.3	66.4	52.4	57.0	59.6	61.2	69.0	63.2	59.7
DeYO	✔	56.3	66.9	53.6	58.9	78.9	55.1	58.1	64.0	52.9	67.0	54.8	58.3	60.6	62.6	70.2	64.5	61.4
ZOA	✗	56.2	65.8	50.3	57.4	78.4	55.3	55.4	63.0	54.5	65.6	52.4	57.5	62.3	61.7	69.0	64.4	60.6
EVA-0 (ours)	✗	58.4	66.5	53.4	**59.4**	78.5	58.7	59.0	**65.4**	56.0	67.3	54.7	**59.3**	64.1	64.1	70.9	**66.3**	**62.6**

Sensitivity to perturbation scale μ. We study the effect of the perturbation scale μ in Table I. EVA-0 is robust to μ over a wide range: for $\mu \in [0.01, 0.1]$, its accuracy varies within 0.5% on ViT-Base (66.2–66.7%) and within 1.4% on ResNet50-GN (44.1–45.5%). This is consistent with Figure 2 (a), where symmetric two-sided probing remains stable across perturbation scales, while one-sided SPSA is more sensitive. Notably, we simply set μ=0.06 by default without tuning. When μ exceeds 0.1, the performance degrades, *e.g.*, to 59.5% and 31.6% at μ=0.2, since a larger perturbation increases the Taylor approximation error, which negatively affects both the update direction estimation and the averaged prediction for inference (*c.f.* Sec. 3.3).

Table I: Sensitivity of the perturbation scale μ on ImageNet-C (severity level 5) regarding **Accuracy (%)** under `single-domain TTA`. We use μ=0.06 by default.

μ	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10	0.15	0.20
ViT-Base	66.5	66.5	66.6	66.7	66.7	66.7	66.7	66.6	66.4	66.2	64.1	59.5
ResNet50-GN	44.8	45.0	45.2	45.4	45.5	45.4	45.3	45.0	44.6	44.1	39.3	31.6

Sensitivity to alignment coefficient λ. We study the effect of the coefficient λ of ℓ_{SWA} in Table J, where λ_0 denotes our default value (λ_0=500 for ViT-Base and λ_0=50 for ResNet50-GN). We choose

λ_0 such that the entropy and alignment terms produce gradients of comparable magnitudes, which requires no labels. EVA-0 is robust to λ over a wide range, achieving over 66.0% on ViT-Base for $\lambda \in [0.4\lambda_0, 2\lambda_0]$ and at least 45.0% on ResNet50-GN for $\lambda \in [0.6\lambda_0, 2\lambda_0]$. As λ decreases further, the performance gradually drops toward the entropy-only variant EVA-0† (λ=0), indicating that sample-wise feature alignment provides a complementary signal to the shortcut-resistant entropy.

Table J: Sensitivity of the alignment coefficient λ on ImageNet-C (severity level 5) regarding **Accuracy (%)** under `single-domain TTA`. λ_0=500 for ViT-Base and λ_0=50 for ResNet50-GN. The default λ_0 is highlighted.

λ	$0.1\lambda_0$	$0.2\lambda_0$	$0.4\lambda_0$	$0.6\lambda_0$	$0.8\lambda_0$	λ_0	$1.2\lambda_0$	$1.4\lambda_0$	$1.6\lambda_0$	$1.8\lambda_0$	$2.0\lambda_0$
ViT-Base	64.7	65.4	66.1	66.5	66.7	66.7	66.7	66.7	66.6	66.5	66.4
ResNet50-GN	42.4	43.4	44.5	45.0	45.3	45.4	45.5	45.5	45.5	45.5	45.5

Effect of output decentering in E^{SR}. Eqn. (6) combines scale normalization with output decentering, which address two different shortcuts of entropy minimization. Scale normalization prevents reducing entropy by inflating the logit norm, yet entropy can still be minimized by collapsing predictions into a few dominant classes, *e.g.*, confidently predicting "cat" for all inputs. Decentering subtracts the online output center c_t, which captures the persistent output bias from historical predictions, so that this bias is not further reinforced during adaptation. To isolate its effect, we optimize all variants with BP under the online imbalanced label shift protocol of SAR [40], where such collapse is most likely. As shown in Table K, making entropy scale-invariant improves TENT from 23.8% to 31.7%, and further subtracting c_t boosts the accuracy to 48.0%, showing that both terms are necessary for shortcut-resistant entropy minimization.

Table K: Effect of output decentering in Eqn. (6). All methods are optimized with BP for a direct comparison. We report average accuracy (%) over 15 corruptions on ImageNet-C (severity level 5) with ResNet50-GN under online imbalanced label shifts (imbalance ratio = ∞).

Method	Scale norm.	Decentering	Acc.
NoAdapt	–	–	31.4
SAR	–	–	33.6
DeYO	–	–	39.4
Entropy (TENT)	✗	✗	23.8
E^{SR} w/o decentering	✔	✗	31.7
E^{SR} (ours)	✔	✔	**48.0**

Sensitivity to random seeds and corruption orders. We evaluate the sensitivity of EVA-0 to perturbation seeds under single-domain TTA and to corruption orders under continual TTA in Table L. EVA-0 consistently achieves the highest accuracy with the lowest standard deviation in both settings. For example, under continual TTA, EVA-0 improves the accuracy of ZOA from 35.1% to 45.6% while reducing the standard deviation from 1.01 to 0.04, *i.e.*, by over 20×. In contrast, BP-based TENT and SAR and the evolution-based FOA vary substantially across corruption orders, *e.g.*, FOA has a standard deviation of 7.44. This insensitivity to perturbation seeds and corruption orders further underscores the reliability of EVA-0 for real-world deployment.

Table L: Sensitivity to random seeds and corruption orders on ImageNet-C (severity level 5) with ResNet50-GN regarding **Accuracy (%)**. We report the mean and standard deviation over 3 random seeds for `single-domain TTA` and 3 different corruption orders for `continual TTA`, following the protocols of Tables 1 and 2.

Setting	TENT	SAR	ZOA	FOZO	FOA	EVA-0 (ours)
Single-domain (3 seeds)	23.4±3.96	37.5±2.21	37.1±0.09	33.7±0.17	35.3±0.09	**45.3**±0.09
Continual (3 orders)	2.0±1.11	36.7±2.55	35.1±1.01	32.3±0.88	18.8±7.44	**45.6**±0.04

More analysis on the entropy shortcut of ZO optimization. The logit vector h can be decomposed into its scale and direction, *i.e.*, $h = \|h\| \cdot h/\|h\|$. Our shortcut refers specifically to the scale component: the model reduces entropy simply by increasing the overall logit magnitude, without changing the relative relationships among classes, which lowers the objective but provides no meaningful task improvement. By contrast, assigning a large logit to the true class and small logits to the others alters the logit direction, and is thus a task-relevant improvement rather than a shortcut. To

further verify that naive ZO entropy minimization tends to exploit this shortcut, besides the entropy and logit norm, we record two additional metrics for each sample and report their averages in Table M:

- *Logit-normalized entropy*: the entropy computed after rescaling logits to the average logit norm of the source model (32.44), which removes the effect of logit scale.
- *Cosine similarity* $\cos(\hat{y}^{\mathrm{src}}, \hat{y}^{\mathrm{tar}})$: the similarity between the source and adapted logits of the same input, which is scale-invariant and mainly reflects changes in logit direction.

Both BP and ZO reduce entropy while increasing the logit norm. However, after controlling for scale, BP still reduces the logit-normalized entropy from 3.33 to 2.65, indicating meaningful directional updates. In contrast, ZO only marginally reduces it to 3.20, and keeps a higher cosine similarity to the source logits (0.79 versus 0.68), *i.e.*, it changes the logit direction less. Together with Figure 1 (a), these results show that naive ZO reduces entropy mainly through logit-scale inflation rather than task-relevant directional updates, which motivates our scale-invariant objective in Sec. 3.1.

Table M: Comparison between BP and ZO optimization under online entropy minimization on ImageNet-C (Gaussian blur, severity level 5) with ViT-Base. Logit-normalized entropy rescales logits to the average logit norm of the source model before computing entropy. $\cos(\hat{y}^{\mathrm{src}}, \hat{y}^{\mathrm{tar}})$ is the cosine similarity between the source and adapted logits of the same input.

Method	Acc. (%)	Entropy	Logit norm	Logit-normalized entropy	$\cos(\hat{y}^{\mathrm{src}}, \hat{y}^{\mathrm{tar}})$
NoAdapt	35.6	3.39	32.44	3.33	1.00
BP optimization	55.3	1.50	40.16	2.65	0.68
ZO optimization	46.7	2.43	37.10	3.20	0.79

More discussions with parameter-dynamics and anchor-based TTA methods. We discuss the differences between EVA-0 and two related BP-based TTA methods, *i.e.*, SIGMA [31] and CMF [32].

Compared with SIGMA: different analysis target, setting, and finding. SIGMA studies the gradient variability of effective BP learning under a Gaussian approximation of parameter transitions, comparing stable and unstable BP adaptation. In the two-forward ZO setting, however, each gradient estimate is a random direction scaled by a scalar, so its variability mainly reflects perturbation noise rather than the underlying dynamics. EVA-0 thus tracks the parameter deviation $\|\theta_t - \theta_0\|$ and compares ZO with BP on the same continual stream. Without the Gaussian assumption, EVA-0 reveals a ZO-specific phenomenon: *random ZO updates accumulate substantially larger drift than BP*, which hinders long-term stability and motivates trajectory control for future ZO-based TTA.

Compared with CMF: different control mechanism–EVA-0 is loss-driven and more memory-efficient. CMF maintains two extra reference states, *i.e.*, a source model and a recursively refined source model, plus a Kalman covariance state, and fuses them with pre-scheduled weights independent of the current adaptation. In contrast, EVA-0 keeps only one anchor for both anchor-guided exploration and weak relaxation, reducing memory overhead. Moreover, it adapts both the anchor-relative direction and step size to the current loss difference, enabling loss-driven trajectory control that moves either toward or away from the anchor. Thus, EVA-0 offers a memory-efficient and loss-driven anchor-control mechanism tailored to two-forward ZO adaptation.

E.1 Limitations, Future Directions, and Broader Impacts

EVA-0 is mainly evaluated on standard vision tasks. Extending it to broader tasks, modalities, and real-world deployment scenarios would be an interesting direction for future work. More broadly, the two-forward, backpropagation-free nature of EVA-0 opens up new opportunities for efficient test-time evolution across diverse model families and deployment settings, including black-box model optimization where only forward access is available.

This work focuses on technical contributions. The proposed method is a fundamental approach that may positively support efficient and adaptive model deployment on resource-limited devices. We do not identify any direct negative societal impacts that require specific discussion.