IS ADAPTATION WORTH THE COST? ECONOMICS-GUIDED TEST-TIME ADAPTATION UNDER BUDGET CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-Time Adaptation (TTA) enables models to dynamically adjust to non-stationary test distributions through online updates during inference. However, when computational resources are limited, existing methods with pre-defined update schedules may waste budget on low-value or unreliable batches, leading to suboptimal overall utility. This raises a critical yet overlooked question: under a limited test-time budget, which adaptations are truly worth performing? To address this issue, we propose **ECO-TTA**, an ECOnomics-guided TTA framework that reformulates adaptation as a cost-aware investment process. Specifically, ECO-TTA defines a multi-level action space and treats each candidate action as a purchasable asset, whose price is its adaptation cost and value is its performance payoff. To estimate this payoff without labels, ECO-TTA derives an efficient value estimator that transfers realized adaptation gains from statistically similar historical batches to predict the expected return of the current asset. It further designs an adaptive budget controller based on the shadow price, which quantifies the opportunity cost of current budget consumption and adjusts the cost penalty along budget consumption trajectories. Together, these designs establish a unified cost-benefit criterion for online decision-making, enabling an adaptive trade-off between limited budgets and high-value updates. Our extensive experiments demonstrate the consistent superiority of ECO-TTA across diverse shift types, TTA scenarios, and budget settings. The code will be publicly available.

1 INTRODUCTION

Deep learning models increasingly serve as crucial components of non-stationary real-world systems, ranging from autonomous perception (Tang et al., 2022; Ma et al., 2023) to robotic interaction (Robinson et al., 2023). In these scenarios, test data rarely follows the same distribution during training, as sensor noise, object appearance, and deployment environments may continuously change (Torralba & Efros, 2011; Koh et al., 2021). These shifts substantially degrade model performance, yet most models remain statically deployed, with parameters frozen throughout inference (Ben-David et al., 2010; Recht et al., 2019; Sun et al., 2020). This static paradigm limits model usability in the open world, where models must remain reliable not on a single benchmark distribution, but over long and evolving test streams.

Test-Time Adaptation (TTA) addresses this limitation by turning inference into an adaptive process (Wang et al., 2020; Zhou & Levine, 2021; Hu et al., 2025). By optimizing self-supervised objectives on incoming batches, TTA provides a practical mechanism for dynamic deployment under distribution shift. Despite its effectiveness, adaptation is not free: each update requires additional forward or backward passes, increasing overhead and processing latency (Zhang et al., 2022; Alfarra et al., 2024). As a result, adaptation cannot be indiscriminately scaled up at test time, especially when computational resources are limited.

This raises a fundamental yet largely overlooked question: under a limited budget, which adaptations are worth performing? Existing TTA studies typically follow a pre-defined schedule, *i.e.*, taking a fixed number of gradient steps at every time step, as shown in Fig. 1(a). This one-size-fits-all strategy implicitly assumes all test batches deserve equal adaptation effort, overlooking the need for global

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

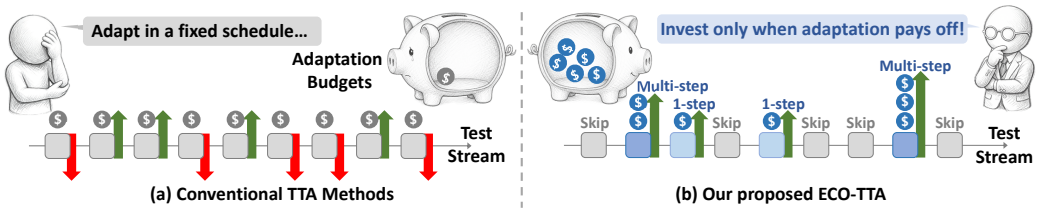


Figure 1: Illustration of different budget allocation strategies in TTA. (a) Prior works: Adaptation under a pre-defined schedule. (b) Ours: Turns adaptation into an online investment process.

allocation. In practical deployment, however, this often fails to maximize overall utility: adapting to low-value batches can waste computation and cause negative transfer, whereas allocating more steps to high-value batches can yield greater gains.

To address this challenge, we propose **ECO-TTA**, an ECOnomics-guided TTA framework that reformulates budget-constrained adaptation as an online investment problem, as shown in Fig. 1(b). Under this view, each candidate action becomes a purchasable asset, priced by computation and valued by its performance payoff. Instead of the equal-value assumption, ECO-TTA evaluates assets via a bidirectional temporal lens. Retrospectively, it builds a historical memory that retrieves past batches with similar statistics and leverages their realized adaptation gains to estimate the current payoff. Prospectively, it introduces a dynamic shadow price to estimate the potential value of budget reserved for future shifts, quantifying the opportunity cost of current adaptation. Together, these designs establish a unified cost-benefit criterion for online decision-making, enabling an adaptive trade-off between limited budgets and high-value updates.

Specifically, ECO-TTA defines a multi-level action space with heterogeneous costs, ranging from a cheap skip operation to costly multi-step updates. Since the true payoff of each action is unobservable without test-time labels, we derive a disagreement-based accuracy proxy to estimate the pre- and post-adaptation performance gap, and store this payoff into the value memory. To generalize past experience to the current batch, we propose a similarity-weighted value estimator that aggregates historical gains for one-step adaptation and extrapolates them to multi-step actions, yielding stable action-specific value predictions. On top of this estimator, an adaptive budget controller updates the shadow price according to the deviation between actual spending and the planned budget pace, translating the global constraint into a time-varying opportunity cost for each action. By interpreting the shadow price as a Lagrange multiplier, we derive an online decision rule that selects the feasible action with the largest net utility, balancing predicted gain and resource limitation.

We evaluate the effectiveness and generalizability of ECO-TTA across diverse shift types, TTA scenarios, and budget settings. Across a range of base TTA methods, ECO-TTA consistently outperforms existing scheduling strategies, yielding an average gain of **+11.18%**. Furthermore, we showcase the strong budget efficiency of our allocation strategy, surpassing full adaptation while using only **10%** of the computation.

Contributions. 1) To the best of our knowledge, we pioneer dynamic resource allocation in online TTA, addressing a critical efficiency gap in real-world deployments. **2)** We propose an efficient value estimator based on a closed-form approximation to accuracy gain, enabling label-free payoff prediction with negligible extra overhead. **3)** We design an adaptive budget controller inspired by the economic concept of shadow price, tracking the budget consumption trajectory to guide online selection of high-value actions. **4)** Experiments across three datasets, four scenarios, and six budget settings demonstrate ECO-TTA's superiority. Notably, our results reveal a key insight: under limited resources, the central question is not merely how to adapt, but when adaptation is worth its cost.

2 Related Work

Test-Time Adaptation aims to improve model performance on out-of-distribution samples during inference (Lim et al., 2023; Yuan et al., 2023; Liang et al., 2025). Depending on whether the source training process is modified, existing methods can be broadly divided into test-time training (Hakim et al., 2023; Liu et al., 2024; Durasov et al., 2025) and fully test-time adaptation (Boudiaf et al., 2022; Tsai et al., 2024; Zhou et al., 2025). In this paper, we focus on fully TTA, as it can be flexibly applied to off-the-shelf models. Fully TTA methods mainly fall into three categories. Entropy-minimization methods improve prediction confidence on shifted samples by optimizing output entropy (Niu et al., 2022; Lee et al., 2024; Hu et al., 2025; Chen et al., 2026a). Statistics-alignment methods reduce

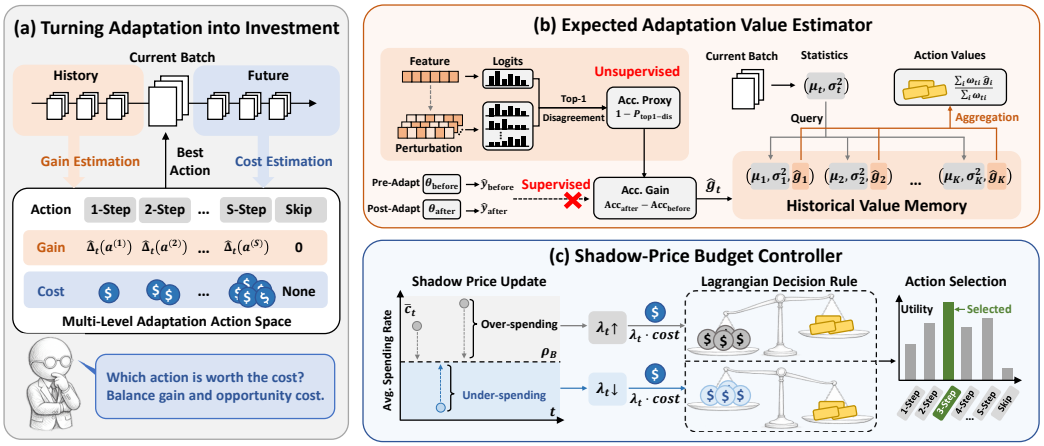


Figure 2: Overview of ECO-TTA. a) We formulate adaptation as online investment, treating each action as an asset selected by its expected gain and opportunity cost. b) ECO-TTA stores historical outcomes in a value memory, providing reference to estimate current batch's payoff. c) A shadow-price controller dynamically adjusts the opportunity cost and selects actions via a Lagrangian rule.

distribution mismatch by aligning batch normalization or feature statistics between source and target domains (Mirza et al., 2023; Hu et al., 2026). Teacher-student methods stabilize adaptation by using teacher predictions, momentum models, or temporal ensembles as pseudo-supervision (Wang et al., 2022; Lee & Chang, 2024; Chen et al., 2026b). Overall, these works primarily design better optimization objectives or stabilization mechanisms, focusing on *how to adapt*.

TTA Evaluation Protocols mainly differ in how they construct test-time distribution shifts. Early benchmarks focus on designing various image-level shifts, such as corruptions (Hendrycks & Dietterich, 2019), texture variations (Wang et al., 2019), and style changes (Hendrycks et al., 2021; Bashkirova et al., 2022). Building on these benchmarks, later works introduce more realistic temporal challenges, including continuously changing, mixed-shift, or non-stationary test streams (Wang et al., 2022; Press et al., 2024; Niu et al., 2023; Du et al., 2024). Other settings restrict the test stream by considering small batches or even batch-size-one evaluation (Zhao et al., 2024; Tomar et al., 2024; Ma & Qiao, 2025). Despite these advances, the time cost of online updates remains an important yet underexplored dimension in TTA evaluation. Existing limited-resource studies highlight this practical concern (Alfarra et al., 2024), but largely focus on measuring performance degradation under constrained budgets. In contrast, this work asks *when adaptation is worth its cost* and further studies how to allocate the available adaptation budget online.

3 Methodology

In this paper, we propose **ECO-TTA**, a budget-aware TTA framework that reformulates adaptation as an online investment process. It incorporates three core designs: a multi-level adaptation action space that defines assets with different update intensities in Fig. 2(a), a historical value estimator that predicts adaptation gain from reusable past experience in Fig. 2(b), and a shadow-price budget controller that dynamically prices the opportunity cost of computation in Fig. 2(c).

3.1 Turning Adaptation into Investment

Existing TTA methods typically follow a fixed adaptation frequency, implicitly assuming equal value across test batches and leading to suboptimal overall utility. To break this assumption, we reformulate TTA as the sequential investment of adaptation assets, moving from uniform per-batch updates to value-aware budget allocation.

Multi-Level Adaptation Action Space. To support flexible adaptation across heterogeneous test batches, we define the update procedure of any base TTA algorithm into a unit action. Built upon this, we construct a multi-level action space that replaces the conventional fixed adaptation with adaptive choices over update intensity and model recovery:

$$\mathcal{A} = \left\{ a^{(s)} \mid s \in \{0, 1, 2 \ldots\} \right\} \cup \left\{ a^{\text{reset}} \right\}, \tag{1}$$

where $a^{(0)}$ skips adaptation and directly performs inference with the current model; $a^{(1)}$ denotes a single update step; $a^{(s)}$, for $s \geq 2$, performs s consecutive steps on the same batch; and a^{reset} restores the source model to mitigate error accumulation caused by prolonged adaptation. To compare these actions under a unified resource scale, we normalize one update step as a unit cost, while assigning the reset operation a method-specific cost according to its actual computational overhead:

$$c(a) = \begin{cases} s, & a = a^{(s)}, \quad s \in \{0, \ldots, S\}, \\ c_{\text{reset}}, & a = a^{\text{reset}}, \end{cases} \tag{2}$$

where $c(a)$ denotes the normalized cost of action a.

Online Investment Problem. Based on the above action space and cost assignment, we cast budget-aware TTA as an online investment process, where each adaptation action is treated as a purchasable asset. Given a sequentially arriving batch stream $\mathcal{X} = \{x_1, \ldots, x_T\}$, the model selects an asset $a_t \in \mathcal{A}$ at each time step according to a decision policy π. Each selected asset consumes $c(a_t)$ and yields an adaptation gain $\Delta_t(a_t; \theta_t, x_t)$, where θ_t denotes the model state before taking action a_t. The objective is to maximize cumulative adaptation utility under a resource budget B:

$$\max_{\pi} \ \mathbb{E}_{\pi} \left[\sum_{t=1}^{T} \Delta_t(a_t; \theta_t, x_t) \right], \quad \text{s.t.} \quad \sum_{t=1}^{T} c(a_t) \leq B, \quad a_t \sim \pi(\cdot \mid \mathcal{H}_t, b_t), \tag{3}$$

where $\mathcal{H}_t$ is the investment trajectory up to time t, and $b_t = B - \sum_{i=1}^{t-1} c(a_i)$ is the remaining budget.

3.2 Expected Adaptation Value Estimator

Under the investment formulation above, the key question becomes: *when is adaptation worth its cost?* Ideally, each asset should be purchased only when its expected payoff outweighs the incurred cost. To realize this, we develop a history-based estimator equipped with an unsupervised gain proxy, serving as the decision basis for online adaptation.

Efficient Disagreement-Based Gain Proxy. Since ground-truth labels are unavailable at test time, the actual accuracy gain after adaptation cannot be directly observed. This prevents the investment trajectory from receiving explicit payoff feedback, making it difficult to evaluate whether a past adaptation was truly beneficial. We therefore seek an unsupervised accuracy estimation of such feedback. Existing methods commonly exploit the statistical correspondence between prediction disagreement and error (Baek et al., 2022; Jiang et al., 2021), approximating error rate by the probability that the top-1 prediction changes under parameter perturbations.

In practice, perturbations are commonly instantiated by Monte Carlo dropout (Srivastava et al., 2014), where different masks induce stochastic subnetworks and disagreement is measured by the probability that the top-1 prediction changes across subnetworks. However, directly estimating this probability requires aggregating predictions over many dropout samples. The resulting computation and memory costs scale nearly linearly with the number of sampling passes, making it unsuitable for real-time TTA. To remove this bottleneck, we analytically characterize the dropout-induced perturbation on pairwise logit gaps and derive a closed-form concentration bound for the disagreement. This converts stochastic estimation into an efficient proxy of unsupervised accuracy.

Proposition 1 (Closed-Form Disagreement Proxy). *Let $z \in \mathbb{R}^d$ be the backbone feature, and let classifier logits be*

$$h = Az + b, \qquad p_i = \text{softmax}(h)_i,$$

where $A \in \mathbb{R}^{C \times d}$, $b \in \mathbb{R}^C$. a_i denotes the i-th row of A. Consider the dropout applied to the feature:

$$\widetilde{z} = \frac{m}{q} \odot z, \qquad m_k \sim \text{Bernoulli}(q),$$

where q is the keep probability. $\widetilde{h}_i = a_i \widetilde{z} + b_i$ is the perturbed logit. Let $c = \arg\max_r h_r$ be the no-dropout top-1 class, with classification margin $\gamma_i = h_c - h_i > 0$ for each $i \neq c$.

Under the Central Limit Assumption, the disagreement probability

$$P_{\text{top1-dis}} = \Pr_{m} \left(\arg\max_{r} \widetilde{h}_r \neq c \right)$$

admits the following upper bound:

$$P_{\text{top1-dis}} \lesssim \sum_{i \neq c} \exp\left(-\frac{q}{2(1-q)} \cdot \frac{\gamma_i^2}{V_i}\right), \quad \textit{where} \quad V_i = \sum_{k=1}^{d} \left[(a_i - a_c)_k z_k\right]^2. \tag{4}$$

Thus, the right-hand side provides a closed-form proxy for dropout-induced top-1 disagreement.

All detailed analysis and proof are provided in Appendix A.

Notably, Proposition 1 yields a deterministic metric for stochastic disagreement, requiring only a single forward pass. Based on this, we estimate adaptation gain by comparing the proxy accuracy before and after adaptation, providing lightweight online payoff feedback:

$$\widehat{g} = \widehat{A}_{\text{after}} - \widehat{A}_{\text{before}}, \quad \widehat{A} = 1 - P_{\text{top1-dis}}. \tag{5}$$

Historical Value Memory. To estimate value before action selection, we maintain a historical memory $\mathcal{M}$ that records past adaptation responses as reusable decision reference. Each historical asset is represented as $\mathcal{E}_i = (\mu_i, \sigma_i^2, \hat{g}_i)$, where μ_i and σ_i^2 denote the batch-level feature mean and variance, and $\hat{g}_i$ is the observed gain derived from Eq. 5. For the current batch x_t, we extract its descriptor (μ_t, σ_t^2) and compare it with historical assets using the ℓ_2 distance between their statistics.

For storage efficiency, $\mathcal{M}$ is maintained as a first-in-first-out queue with maximum size of $K = 50$. Retrospectively, we estimate the one-step adaptation value by a similarity-weighted aggregation of historical gains, and extrapolate it to multi-step actions with a diminishing-return prior:

$$\widehat{\Delta}_t(a^{(1)}) = \frac{\sum_{i \in \mathcal{M}} \omega_{ti} \hat{g}_i}{\sum_{i \in \mathcal{M}} \omega_{ti}}, \qquad \widehat{\Delta}_t(a^{(s)}) = \frac{1 - e^{-\beta s}}{1 - e^{-\beta}} \widehat{\Delta}_t(a^{(1)}), \quad s \in \{2, \ldots, S\}, \tag{6}$$

where ω_{ti} is the similarity weight to the i-th memory asset, and β controls the saturation of additional steps. We set $\beta = 0.5$ by default based on empirical observations.

For the reset action, we follow the reset indicator used in SAR (Niu et al., 2023) to detect model collapse. When the indicator is activated, we assign $\widehat{\Delta}_t(a^{\text{reset}})$ a sufficiently large value to prioritize model recovery; otherwise, its value is set to a sufficiently small value to avoid unnecessary resets.

After each executed adaptation, the observed gain by Eq. 5 is written back to $\mathcal{M}$, progressively turning historical adaptations into reusable decision assets. In this way, the estimator closes the loop between before-adapt value prediction and after-adapt value record: past adaptation outcomes provide empirical payoff evidence, while the current batch retrieves the relevant evidence through feature-statistic similarity.

3.3 Shadow-Price Budget Controller

Beyond value estimation, budget-aware TTA must also regulate the spending pace over the test stream. A high-value adaptation can still be suboptimal if it consumes resources that may be more valuable for harder future batches. We therefore develop a budget controller that balances immediate adaptation gain against future budget availability.

Dynamic Shadow Price Update. Since the test stream is online, the static cost in Eq. 2 cannot reliably match the evolving budget schedule. Inspired by the economic notion of shadow price (Heckman, 1974; Drèze & Stern, 1990), we introduce λ_t as an adaptive regulation factor that reflects the opportunity cost induced by the current scarcity of the remaining budget. We update it according to the deviation between cumulative spending and the planned budget trajectory:

$$\lambda_{t+1} \leftarrow \left[\lambda_t + \eta \left(\Sigma_{i=1}^{t} c(a_i)/t - \rho_B\right)\right]_+, \ \ \rho_B = \frac{B}{T}, \tag{7}$$

where η is the update rate. $[\cdot]_+$ ensures the non-negativity.

When the cumulative cost exceeds the planned budget, λ_t increases, penalizing costly actions and making future adaptation more selective. Conversely, when the policy spends below schedule, λ_t decreases, relaxing the cost penalty and allowing aggressive adaptation when beneficial. In this way, the global budget constraint is converted into an adaptive step-wise opportunity cost.

Table 1: Experimental results on ImageNet-C (severity level 5) under three wild settings with the **Global** budget protocol. We report accuracy (%) averaged over three independent runs. The better result in each comparison is highlighted in **bold**.

Budget Ratio $r = 0.1$													
Method	Reference	Scheduler	Label Shift					Batch Size = 1					Mix Shift
			Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg	Avg
TENT	ICLR'21	Uniform	30.12	40.47	52.74	53.38	45.11	26.85	37.72	50.58	50.88	42.48	49.68
		Ours	**54.49**	**52.72**	**64.63**	**61.37**	**58.56**	**53.31**	**51.20**	**62.73**	**60.16**	**57.09**	**57.41**
SAR	ICLR'23	Uniform	32.18	42.01	54.40	53.76	46.48	18.25	33.14	42.90	47.59	36.62	50.37
		Ours	**56.24**	**55.60**	**66.47**	**63.51**	**60.74**	**55.12**	**51.76**	**61.58**	**60.41**	**57.36**	**60.17**
ActMAD	CVPR'23	Uniform	0.12	2.02	0.57	2.38	1.35	0.28	0.62	0.74	0.72	0.61	1.32
		Ours	**7.24**	**6.56**	**12.53**	**12.36**	**9.83**	**7.12**	**5.85**	**15.82**	**12.03**	**10.41**	**29.73**
CMF	ICLR'24	Uniform	50.06	57.28	69.93	68.14	62.11	16.62	30.42	39.33	44.10	33.68	58.63
		Ours	**60.69**	**61.16**	**73.67**	**69.54**	**66.64**	**51.19**	**45.01**	**61.07**	**53.97**	**52.92**	**61.36**
ReCAP	ICML'25	Uniform	43.79	50.70	63.57	62.38	55.86	27.41	41.50	55.45	54.89	45.97	56.20
		Ours	**57.60**	**54.09**	**68.43**	**65.24**	**61.59**	**59.98**	**58.59**	**69.99**	**67.47**	**64.28**	**60.56**
ZeroSiam	ICLR'26	Uniform	31.37	41.22	54.41	54.39	46.28	27.04	38.03	50.73	51.19	42.73	50.46
		Ours	**56.19**	**54.71**	**66.90**	**63.31**	**60.55**	**55.49**	**52.74**	**65.41**	**62.87**	**59.37**	**59.88**
Budget Ratio $r = 0.5$													
Method	Reference	Scheduler	Label Shift					Batch Size = 1					Mix Shift
			Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg	Avg
TENT	ICLR'21	Uniform	43.73	51.15	45.28	61.71	50.92	39.26	47.76	49.03	58.53	49.27	34.52
		Ours	**57.33**	**56.34**	**67.45**	**65.22**	**61.87**	**56.82**	**55.83**	**65.87**	**64.26**	**60.95**	**58.59**
SAR	ICLR'23	Uniform	45.87	52.63	56.98	62.92	55.18	29.74	43.55	57.12	56.68	47.91	56.05
		Ours	**59.54**	**58.56**	**69.88**	**67.59**	**64.18**	**58.62**	**57.30**	**67.76**	**64.79**	**62.35**	**61.10**
ActMAD	CVPR'23	Uniform	0.10	1.27	0.43	0.87	0.71	0.10	0.10	0.10	0.10	0.10	2.81
		Ours	**7.83**	**6.53**	**11.46**	**9.98**	**9.02**	**10.95**	**5.96**	**16.09**	**11.94**	**11.25**	**29.60**
CMF	ICLR'24	Uniform	71.93	74.62	82.09	79.72	77.43	17.15	30.97	40.68	45.06	34.55	60.29
		Ours	**74.98**	**75.41**	**82.98**	**80.70**	**78.75**	**51.66**	**44.82**	**60.33**	**55.47**	**53.16**	**61.42**
ReCAP	ICML'25	Uniform	53.19	58.77	70.36	68.79	63.42	44.54	53.54	66.07	65.02	58.14	59.35
		Ours	**60.70**	**60.56**	**72.27**	**69.96**	**66.22**	**61.86**	**62.28**	**72.81**	**71.02**	**67.33**	**62.03**
ZeroSiam	ICLR'26	Uniform	45.25	52.50	64.59	63.08	57.10	40.65	48.45	61.62	59.43	53.33	55.13
		Ours	**59.59**	**58.66**	**70.00**	**67.72**	**64.29**	**58.91**	**57.26**	**68.99**	**66.76**	**63.25**	**60.69**

Online Lagrangian Decision Rule. Given the adaptive opportunity cost encoded by the shadow price, we solve the online investment problem through a Lagrangian relaxation. Specifically, the hard budget constraint is converted into a cost penalty weighted by the current shadow price λ_t. This yields a net utility score for each candidate action:

$$U_t(a; \lambda_t) = \widehat{\Delta}_t(a) - \lambda_t c(a), \tag{8}$$

where $\widehat{\Delta}_t(a)$ is the estimated value in Eq. 6 and $\lambda_t c(a)$ represents the opportunity cost in Eq. 7.

At each step, the controller first filters out actions that exceed the remaining budget. Among the feasible candidates, it selects the action with the largest Lagrangian utility:

$$\mathcal{A}_t^{\text{feas}} = \{a \in \mathcal{A} \,|\, c(a) \leq b_t\}\,, \;\; a_t = \arg\max_{a \in \mathcal{A}_t^{\text{feas}}} U_t(a; \lambda_t). \tag{9}$$

This derived rule forms a self-balancing decision mechanism between expected payoff and budget pressure. Even when the value estimator predicts a high return, the controller may suppress the action if the remaining budget is scarce and its opportunity cost is high. Conversely, when the predicted value is low, abundant remaining budget lowers the shadow price, allowing the controller to spend otherwise idle resources on exploratory adaptation. Thus, adaptation is not triggered by value alone, nor blocked by cost alone; it emerges from their dynamic equilibrium, where computation is allocated when the expected return is justified under the current budget state.

4 EXPERIMENTS

For a fair comparison, we follow the evaluation pipeline of prior works (Niu et al., 2023; Hu et al., 2025), including the datasets, pre-trained models, and training recipes. The only modification lies in the limited-budget protocol, where the original fixed update schedule is compared with ours.

Table 2: Experimental results on ImageNet-C under three wild settings with the **Sliding-Window** budget protocol at ratio 0.2. We report accuracy (%) averaged over three independent runs.

Method	Reference	Scheduler	Label Shift					Batch Size = 1					Mix Shift
			Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg	Avg
TENT	ICLR'21	Uniform	36.74	45.44	54.47	57.25	49.26	31.54	41.44	53.54	53.97	46.03	52.53
		Ours	**54.41**	**51.74**	**64.66**	**61.87**	**58.42**	**52.86**	**49.57**	**62.24**	**58.80**	**56.07**	**57.06**
SAR	ICLR'23	Uniform	39.35	46.63	58.40	57.55	51.22	21.07	36.63	48.81	51.15	40.64	53.42
		Ours	**56.71**	**54.54**	**66.83**	**63.82**	**60.73**	**54.23**	**50.71**	**63.78**	**59.50**	**57.24**	**59.56**
ActMAD	CVPR'23	Uniform	0.25	2.11	0.96	3.21	1.72	0.17	0.13	0.67	0.37	0.35	6.19
		Ours	**5.70**	**3.83**	**9.50**	**16.71**	**9.15**	**6.53**	**7.79**	**18.09**	**11.02**	**11.15**	**28.16**
CMF	ICLR'24	Uniform	58.26	62.01	71.51	68.32	65.48	16.68	30.63	39.76	44.63	34.01	59.62
		Ours	**62.72**	**62.38**	**73.45**	**70.43**	**67.55**	**51.10**	**44.03**	**60.52**	**54.89**	**52.74**	**61.22**
ReCAP	ICML'25	Uniform	48.50	54.77	67.36	65.47	59.73	34.31	46.35	60.27	59.13	51.06	57.87
		Ours	**57.43**	**55.55**	**69.45**	**66.01**	**62.42**	**58.76**	**57.03**	**68.73**	**65.82**	**62.84**	**59.58**
ZeroSiam	ICLR'26	Uniform	38.43	46.69	59.44	58.03	51.46	32.63	42.05	55.75	54.87	47.24	53.03
		Ours	**56.05**	**53.69**	**66.73**	**63.56**	**60.27**	**55.05**	**51.86**	**64.78**	**60.80**	**58.33**	**59.30**

4.1 Experimental Setup

Datasets and TTA Scenarios. We conduct our experiments on three datasets: 1) ImageNet-C (Hendrycks & Dietterich, 2019), a large-scale dataset containing 15 corruption types and 5 severity levels. 2) ImageNet-R (Hendrycks et al., 2021) and 3) ImageNet-S (Wang et al., 2019), two datasets which introduce domain shifts caused by variations in styles and textures (e.g., sketch, cartoon). For TTA scenarios, we follow the mild setting of TENT (Wang et al., 2020) and the three wild settings introduced by SAR (Niu et al., 2023). Further details are provided in Appendix E.

Budget Protocols. To mimic resource-constrained deployment, we evaluate two budget protocols that capture **long-term** and **short-term** computation limits. Here, r denotes the budget ratio relative to the full update schedule, *e.g.*, $r = 0.1$ allows only 10% of its total adaptation cost. Under the *global* protocol, a single budget is shared across the entire test stream, modeling a fixed total resource quota. Under the stricter *sliding-window* protocol, each 100-batch window is assigned the same budget ratio r, modeling short-term throughput constraints and preventing budget saved in one part of the stream from being spent much later.

Compared Methods. We compare ECO-TTA against two limited-budget scheduling baselines: *Uniform*, which evenly distributes the adaptation budget over the test stream, and *Greedy*, which spends the budget as early as possible. Due to space limits, we provide an intuitive visualization of the allocation process in Appendix D.3 and report the Greedy results in Appendix B.2. To verify the generality, we apply scheduler to representative base TTA methods from three categories: entropy-minimization methods, including TENT (Wang et al., 2020), SAR (Niu et al., 2023), ReCAP (Hu et al., 2025), and ZeroSiam (Chen et al., 2026a); statistics-alignment methods, represented by ActMAD (Mirza et al., 2023); and teacher-student-like methods, represented by CMF (Lee & Chang, 2024). By changing only the scheduling strategy while keeping the base TTA method fixed, we isolate the effect of resource allocation from the underlying adaptation algorithm.

Implementation Details. Following standard practice (Niu et al., 2023; Hu et al., 2025), we conduct experiments on ViT-Base (Dosovitskiy et al., 2020) and ResNet-50 (He et al., 2016). ECO-TTA is architecture-agnostic and uses the same configuration across all base models and TTA methods. For optimizer, we use SGD with batch size 64 (except for batch size$= 1$), momentum 0.9, and learning rates 2.5×10^{-4} and 1×10^{-3} for ResNet and ViT. For ECO-TTA, we set the saturation rate $\beta = 0.5$, dropout keep probability $q = 0.5$, shadow-price update rate $\eta = 5 \times 10^{-3}$, and memory bank size $K = 50$ by default. All experiments can be conducted on a single NVIDIA RTX 4090 GPU.

4.2 Main Results

Due to space limitations, the following results in the main paper are reported on ViT. Evaluation on ResNet is provided in the Appendix. B.3 and validates the consistent observation.

Evaluation on Global Budget. We first evaluate ECO-TTA under global budget $r = \{0.1, 0.5\}$ across three wild settings. As shown in Tab. 1, the results yield four key observations. 1) *Superior Scheduling Gain.* Without modifying base methods, ECO-TTA achieves substantial gains purely through adaptive scheduling, outperforming uniform allocation by **+12.36%** and **+9.99%** at $r = 0.1$

and $r = 0.5$. 2) *Low-Data Robustness.* Even in the batch-size-one setting, where both data and computation are limited, ECO-TTA yields an average improvement of $+14.53\%$ across all methods. 3) *Suppression of Negative Adaptation.* More computation can be harmful: under mixed shifts, increasing the uniform budget from 0.1 to 0.5 reduces TENT accuracy by approximately 15%. In contrast, when ActMAD collapses to near-zero accuracy, ECO-TTA suppresses harmful updates and improves its accuracy by more than $\mathbf{10}\times$. 4) *Narrowing Performance Gaps.* With ECO-TTA, the classical TENT performs comparably to the more recent ZeroSiam, revealing a key insight: under limited resources, the central question is not only *how* to adapt, but also deciding *when* to adapt.

Evaluation on Sliding-Window Budget. To simulate short-term constraints in edge deployment, we test ECO-TTA under a sliding-window budget with $r = 0.2$. As shown in Tab. 2, the results consistently support the observations from the global-budget setting and validate ECO-TTA outperforms uniform scheduler across all methods and scenarios, achieving an average gain of $+8.94\%$. This shows that ECO-TTA does not rely on long-horizon budget accumulation, but can make flexible allocation decisions within local windows.

Evaluation on More Scenarios. We further evaluate ECO-TTA on ImageNet-R and ImageNet-S under the mild TTA setting to test generalization beyond synthetic corruptions. As shown in Tab. 3, the results yield several observations. 1) *Gains under Style Shifts.* ECO-TTA improves average accuracy by $+10.56\%$ on ImageNet-R, showing its effectiveness under style shifts. 2) *Robustness to Severe Shifts.* The gain is even larger on ImageNet-S, where color and texture cues are largely removed, reaching **+17.70%**. 3) *Broad Applicability.* Together with Tabs. 1, 2, and 3, ECO-TTA consistently improves performance across both mild and wild TTA settings.

Table 3: Experimental results on ImageNet-R and ImageNet-S under mild TTA settings.

Method	Scheduler	ImageNet-R			ImageNet-S		
		Window	Global	Avg.	Window	Global	Avg.
TENT	Uniform	46.56	48.03	47.30	23.83	13.15	18.49
	Ours	**56.58**	**57.51**	**57.05**	**43.32**	**44.91**	**44.12**
SAR	Uniform	46.59	49.33	47.96	28.32	31.01	29.67
	Ours	**58.57**	**60.24**	**59.41**	**45.34**	**47.15**	**46.25**
ReCAP	Uniform	53.66	57.87	55.77	35.86	40.39	38.13
	Ours	**63.73**	**67.11**	**65.42**	**49.16**	**51.75**	**50.46**
ZeroSiam	Uniform	46.58	49.83	48.21	28.74	32.70	30.72
	Ours	**58.85**	**60.33**	**59.59**	**45.91**	**48.03**	**46.97**

Table 4: Comparison of ECO-TTA under global budget ratios $r \in \{0.1, 0.5, 1.0\}$ and Full adaptation (*i.e.*, uniform scheduling with $r = 1.0$). The best and second-best results in each row are shown in **bold** and underlined, respectively.

Method	ImageNet-C				ImageNet-R			
	$r = 0.1$	0.5	1.0	Full	$r = 0.1$	0.5	1.0	Full
TENT	55.41	59.39	**61.07**	49.28	56.01	57.51	**59.46**	44.39
SAR	58.04	62.03	**64.07**	54.32	57.77	60.24	**61.96**	51.66
ReCAP	64.25	67.00	**67.96**	64.21	62.69	67.11	**68.42**	60.92
ZeroSiam	59.95	61.91	**63.84**	59.93	58.00	60.33	**62.44**	54.08

Comparison with Full Adaptation. Tab. 4 compares ECO-TTA with full adaptation, *i.e.*, updates every batch. ECO-TTA exhibits a clear **less-is-more** effect: With only 10% of the budget, ECO-TTA already surpasses full adaptation across all methods on both ImageNet-C and ImageNet-R. Moreover, progressively increasing the budget to 1.0 consistently improves ECO-TTA, showing that additional computation is beneficial when allocated to valuable updates. These results confirm that ECO-TTA is not merely a cost-saving strategy, but a value-aware scheduler that avoids low-value or harmful adaptation.

Full results for $r \in \{0.1, 0.2, 0.5\}$ under both protocols are provided in Appendix B.1.

5 Ablation Study and Visualization

In this section, without loss of generality, all ablations and visualizations use ReCAP as the base TTA method under the label-shift setting with the global-budget protocol for the sake of brevity. Focusing on pivotal components of ECO-TTA, we perform various experiments to analyze their impacts. Refer to Appendix C for more experiments and analysis.

5.1 Approximation Effect of Proxy

To validate the derived disagreement proxy in Proposition 1, we compare the estimated accuracy $\widehat{A}$ with the ground-truth accuracy computed post hoc. As shown in Fig. 3, the proxy closely tracks the temporal trend of the real accuracy over online batches, with Pearson correlations of 0.862, 0.897, and 0.882 for TENT, CMF, and ReCAP. These strong correlations demonstrate that our closed-form proxy provides stable, label-free payoff feedback for historical value estimation, enabling ECO-TTA to make reliable online decisions without ground-truth labels or costly Monte Carlo sampling.

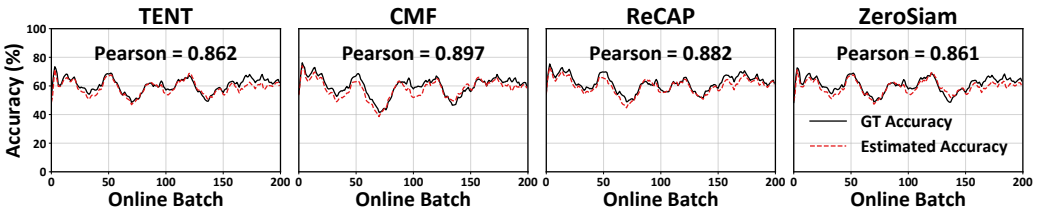


Figure 3: Ground-truth accuracy versus proxy-estimated accuracy during the adaptation process.

5.2 Sensitivity Analysis of Hyperparameters

We assess the robustness of ECO-TTA with respect to two key configurations of value estimator: the memory size K and the saturation rate β. We ablate two coefficients independently.

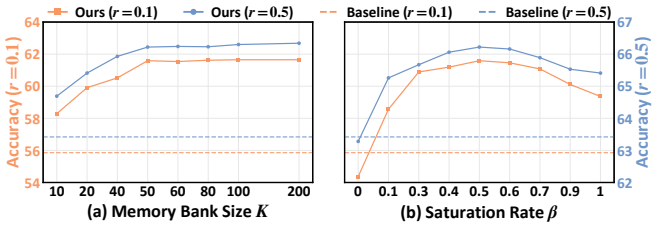


Figure 4: Performance with (a) varying size K of the memory bank, (b) different saturation rate β of additional steps.

As shown in Fig. 4(a), increasing the memory size improves performance under both budget settings, since a larger memory can retain more historical adaptation experience. However, once K exceeds a moderate range (*e.g.*, 50), further enlargement brings only marginal gains. We therefore set 50 by default to balance estimation capacity and efficiency. The saturation rate β controls the diminishing return of additional adaptation steps, and $\beta = 0.5$ matches the average decay trend observed in adaptation gains. As shown in Fig. 4(b), performance peaks around this value. Moreover, ECO-TTA consistently outperforms the baseline across $\beta \in [0.1, 1.0]$, demonstrating its robustness to hyperparameter choices.

5.3 Effectiveness of Components

We investigate the impact of individual components by comparing the full method with partial variants, as shown in Tab. 5. The results provide three key observations. 1) Replacing our derived proxy with prediction entropy substantially degrades performance, indicating that closed-form disagreement provides more reliable payoff feedback than widely-adopted entropy. 2) The shadow-price update improves allocation by adapting the opportunity cost to the spending trajectory. With our proxy but without multi-step actions, adding λ_t raises average accuracy from 55.1% to 59.5% under $r = 0.1$, confirming the need for online budget regulation. 3) Combining all three components achieves the best results across corruption groups and budget ratios, demonstrating their complementary roles in resource-limited TTA.

Table 5: Impact of component variations. $\mathcal{A}$ indicates whether multi-step action is enabled, Proxy indicates whether our derived disagreement proxy or prediction entropy is used, and λ_t indicates whether the dynamic price update is applied.

Base.	$\mathcal{A}$	Proxy	λ_t	Budget Ratio $r = 0.1$					Budget Ratio $r = 0.5$				
				Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg
✔				43.8	50.7	63.6	62.4	55.9	53.2	58.8	70.4	68.8	63.4
	✔		✔	41.8	48.7	61.6	60.4	53.9	52.2	57.8	69.4	67.8	62.4
		✔		43.0	49.9	62.8	61.6	55.1	52.8	58.4	70.0	68.4	63.1
	✔	✔		46.3	51.3	64.5	62.9	56.9	53.4	58.8	70.4	68.8	63.5
		✔	✔	55.4	52.0	66.4	63.2	59.5	60.0	59.5	71.6	69.2	65.4
	✔	✔	✔	**57.6**	**54.1**	**68.4**	**65.2**	**61.6**	**60.7**	**60.6**	**72.3**	**70.0**	**66.2**

6 Conclusion

In this paper, we propose ECO-TTA, a budget-aware framework that transforms test-time adaptation into a closed-loop online investment process. Instead of following a fixed update schedule, ECO-TTA treats adaptation operations as purchasable assets and allocates computation according to their expected payoff and opportunity cost. We introduce a multi-level action space to support flexible adaptation intensities, a history-based value estimator with an efficient disagreement-based gain proxy to predict adaptation payoff, and a shadow-price budget controller to regulate online resource consumption. Extensive experiments across corruption, style, and texture shifts demonstrate that ECO-TTA consistently improves diverse base methods under both global and sliding-window budgets, especially in low-resource and wild test scenarios. We hope this work inspires future research on cost-aware and value-driven adaptation, paving the way for more efficient and reliable deployment under real-world resource constraints.

AI USE STATEMENT

We use LLM as an assistant to help us polish our writing. They were not used to generate experimental data or reported results. All AI-assisted edits were reviewed, revised, and verified by the authors, who take full responsibility for the final content of this work. We support the transparent and responsible use of LLMs in research writing.

ETHICS STATEMENT

This work uses publicly available image-classification benchmarks, including ImageNet-C, ImageNet-R, and ImageNet-S, in accordance with their applicable licenses and terms of use. It involves no human participants or new data collection, and the evaluated models neither use demographic attributes nor make decisions about individuals. We therefore identify no direct human-subject or privacy risks specific to the experimental protocol. The authors declare no financial, professional, or personal conflicts of interest related to this work. We have reviewed and followed the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, the main paper and appendix document the method, theoretical derivations, experimental protocols, and implementation settings. Specifically, Appendix D.4 provides pseudocode for the ECO-TTA allocation procedure; Sec. 4.1 reports the experimental setup and hyperparameter configurations; Appendix E details the datasets and TTA scenarios; Appendix D describes the backbones, base TTA methods, and implementation details; and Appendix A provides complete proofs of the theoretical results.

REFERENCES

Motasem Alfarra, Hani Itani, Alejandro Pardo, shyma yaser alhuwaider, Merey Ramazanova, Juan Camilo Perez, zhipeng cai, Matthias Müller, and Bernard Ghanem. Evaluation of test-time adaptation under computational time constraints. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=6FtAXU4ean.

Christina Baek, Yiding Jiang, Aditi Raghunathan, and J Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. *Advances in Neural Information Processing Systems*, 35:19274–19289, 2022.

Dina Bashkirova, Dan Hendrycks, Donghyun Kim, Haojin Liao, Samarth Mishra, Chandramouli Rajagopalan, Kate Saenko, Kuniaki Saito, Burhan Ul Tayyab, Piotr Teterwak, et al. Visda-2021 competition: Universal domain adaptation to improve performance on out-of-distribution data. In *NeurIPS 2021 Competitions and Demonstrations Track*, pp. 66–79. PMLR, 2022.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

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

Guohao Chen, Shuaicheng Niu, Deyu Chen, Jiahao Yang, Zitian Zhang, Mingkui Tan, Pengcheng Wu, and Zhiqi Shen. Zerosiam: An efficient asymmetry for test-time entropy optimization without collapse. In *Third Workshop on Test-Time Updates (Main Track)*, 2026a.

Xiao Chen, Jiazhen Huang, Zhiming Liu, Qinting Jiang, Fanding Huang, Jingyan Jiang, and Zhi Wang. Test-time distillation for continual model adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 7593–7604, June 2026b.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Jean Drèze and Nicholas Stern. Policy reform, shadow prices, and market prices. *Journal of public economics*, 42(1):1–45, 1990.

Chaoqun Du, Yulin Wang, Jiayi Guo, Yizeng Han, Jie Zhou, and Gao Huang. Unitta: Unified benchmark and versatile framework towards realistic test-time adaptation. *arXiv preprint arXiv:2407.20080*, 2024.

Nikita Durasov, Assaf Shocher, Doruk Oner, Gal Chechik, Alexei A Efros, and Pascal Fua. IT^3: Idempotent test-time training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 14867–14883. PMLR, 13–19 Jul 2025.

Gustavo A Vargas Hakim, David Osowiechi, Mehrdad Noori, Milad Cheraghalikhani, Ali Bahri, Ismail Ben Ayed, and Christian Desrosiers. Clust3: Information invariant test-time training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6136–6145, 2023.

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

James Heckman. Shadow prices, market wages, and labor supply. *Econometrica: journal of the econometric society*, pp. 679–694, 1974.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

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

Zixuan Hu, Yichun Hu, Xiaotong Li, Shixiang Tang, and Ling-Yu Duan. Beyond entropy: Region confidence proxy for wild test-time adaptation. *arXiv preprint arXiv:2505.20704*, 2025.

Zixuan Hu, Xuantuo Huang, Yancheng Li, Yichun Hu, Shengyong Xu, and Ling-Yu Duan. Turning adaptation into assets: Cross-domain bridging for online vision-language navigation. *arXiv preprint arXiv:2605.23257*, 2026.

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of sgd via disagreement. *arXiv preprint arXiv:2106.13799*, 2021.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pp. 5637–5664. PMLR, 2021.

Jae-Hong Lee and Joon-Hyuk Chang. Continual momentum filtering on parameter space for online test-time adaptation. In *The twelfth international conference on learning representations*, 2024.

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

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

Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. Ttn: A domain-shift aware batch normalization in test-time adaptation. *arXiv preprint arXiv:2302.05155*, 2023.

Weihuang Liu, Xi Shen, Haolun Li, Xiuli Bi, Bo Liu, Chi-Man Pun, and Xiaodong Cun. Depth-aware test-time training for zero-shot video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19218–19227, 2024.

Tianyi Ma and Maoying Qiao. Ebar: Efficient buffer and resetting for single-sample continual test-time adaptation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 151–160, 2025.

Xinzhu Ma, Wanli Ouyang, Andrea Simonelli, and Elisa Ricci. 3d object detection from images for autonomous driving: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3537–3556, 2023.

Muhammad Jehanzeb Mirza, Pol Jané Soneira, Wei Lin, Mateusz Kozinski, Horst Possegger, and Horst Bischof. Actmad: Activation matching to align distributions for test-time-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24152–24161, 2023.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023.

Ori Press, Steffen Schneider, Matthias Kümmerer, and Matthias Bethge. Rdumb: A simple approach that questions our progress in continual test-time adaptation. *Advances in Neural Information Processing Systems*, 36, 2024.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Nicole Robinson, Brendan Tidd, Dylan Campbell, Dana Kulić, and Peter Corke. Robotic vision for human-robot interaction and collaboration: A survey and systematic review. *ACM Transactions on Human-Robot Interaction*, 12(1):1–66, 2023.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

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

Yang Tang, Chaoqiang Zhao, Jianrui Wang, Chongzhen Zhang, Qiyu Sun, Wei Xing Zheng, Wenli Du, Feng Qian, and Juergen Kurths. Perception and navigation in autonomous systems in the era of learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12): 9604–9624, 2022.

Devavrat Tomar, Guillaume Vray, Jean-Philippe Thiran, and Behzad Bozorgtabar. Un-mixing test-time normalization statistics: Combatting label temporal correlation. In *International Conference on Learning Representations*, volume 2024, pp. 50967–50988, 2024.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528. IEEE, 2011.

Yun-Yun Tsai, Fu-Chen Chen, Albert YC Chen, Junfeng Yang, Che-Chun Su, Min Sun, and Cheng-Hao Kuo. Gda: Generalized diffusion for robust test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23242–23251, 2024.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2020.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 32, 2019.

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

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

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

Shiji Zhao, Shao-Yuan Li, and Sheng-Jun Huang. Nanoadapt: mitigating negative transfer in test time adaptation with extremely small batch sizes. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 5572–5580, 2024.

Aurick Zhou and Sergey Levine. Bayesian adaptation for covariate shift. *Advances in neural information processing systems*, 34:914–927, 2021.

Zhi Zhou, Kun-Yang Yu, Lan-Zhe Guo, and Yu-Feng Li. Fully test-time adaptation for tabular data. In *Proceedings of the 39th AAAI conference on Artificial Intelligence*, 2025.

Is Adaptation Worth the Cost? Economics-Guided Test-Time Adaptation under Budget Constraints

————Appendix————

The structure of Appendix is as follows:

- Appendix A contains all proofs and propositions for theoretical claims in the main manuscript.
- Appendix B provides additional experimental results across different budget protocols and model architectures, together with comparisons against the greedy scheduler.
- Appendix C provides extended ablation studies on hyper-parameter sensitivity.
- Appendix D outlines further implementation details of our method and baselines.
- Appendix E details the datasets and evaluation protocols used in our experiments.

A Theoretical Proof

Below, we will provide detailed proofs of the theoretical results in the main manuscript.

Notation. We first recall the notation used in the main paper and this appendix. Let C and d denote the number of classes and the feature dimension, respectively. For a test sample, let $z = [z_1, \ldots, z_d]^\top \in \mathbb{R}^d$ be the backbone feature. The classifier is an affine mapping $h = Az + b$, where $A \in \mathbb{R}^{C\times d}$, $b \in \mathbb{R}^C$, $a_i \in \mathbb{R}^{1\times d}$ is the i-th row of A, and b_i is the i-th entry of b. The predicted probability is $p_i = \mathrm{softmax}(h)_i$. Since softmax preserves logit ordering, top-1 disagreement can be analyzed directly in logit space. We denote the no-dropout top-1 class by $c = \arg\max_r h_r$, and define the deterministic margin between class c and each competing class $i \neq c$ as $\gamma_i = h_c - h_i > 0$.

Proposition 1 (Closed-Form Disagreement Proxy). *Let $z \in \mathbb{R}^d$ be the backbone feature, and let classifier logits be*

$$h = Az + b, \qquad p_i = \mathrm{softmax}(h)_i,$$

where $A \in \mathbb{R}^{C\times d}$, $b \in \mathbb{R}^C$, and a_i denotes the i-th row of A. Consider inverted dropout applied to the feature:

$$\widetilde{z} = \frac{m}{q} \odot z, \qquad m_k \sim \mathrm{Bernoulli}(q),$$

where q is the keep probability. $\widetilde{h}_i = a_i\widetilde{z} + b_i$ is the perturbed logit. Let $c = \arg\max_r h_r$ be the no-dropout top-1 class, with classification margin $\gamma_i = h_c - h_i > 0$ for each $i \neq c$.

Under the Central Limit Assumption, the disagreement probability

$$P_{\text{top1-dis}} = \Pr_m\left(\arg\max_r \widetilde{h}_r \neq c\right)$$

admits the following upper bound:

$$P_{\text{top1-dis}} \lesssim \sum_{i\neq c} \exp\left(-\frac{q}{2(1-q)} \cdot \frac{\gamma_i^2}{V_i}\right), \quad \textit{where} \quad V_i = \sum_{k=1}^{d} \left[(a_i - a_c)_k z_k\right]^2. \tag{10}$$

V_i serves as the dropout-induced sensitivity of the logit gap between class i and the deterministic top-1 class c.

Proof. We first rewrite the top-1 disagreement event into a union of pairwise comparison events.

Reduction to Pairwise Logit-Crossing Events. The randomness in $P_{\text{top1-dis}}$ comes only from the dropout mask m. Without dropout, the predicted class is fixed as $c = \arg\max_r h_r$. After applying dropout, the prediction changes from c to another class if some competing class $i \neq c$ attains a perturbed logit no smaller than that of class c. In other words, the original multi-class disagreement

event can be controlled by checking whether any pairwise logit gap between c and a competing class i is reversed.

Formally, define the pairwise crossing event

$$\mathcal{E}_i = \left\{ \widetilde{h}_i \geq \widetilde{h}_c \right\}, \quad i \neq c. \tag{11}$$

If the top-1 prediction under dropout is no longer c, then there must exist at least one class $i \neq c$ whose perturbed logit catches up with or exceeds $\widetilde{h}_c$. Therefore, the disagreement event satisfies

$$\left\{ \arg\max_r \widetilde{h}_r \neq c \right\} \subseteq \bigcup_{i \neq c} \mathcal{E}_i. \tag{12}$$

Consequently, we have the following inequality

$$P_{\text{top1-dis}} = \Pr_m \left(\arg\max_r \widetilde{h}_r \neq c \right) \leq \Pr_m \left(\bigcup_{i \neq c} \left\{ \widetilde{h}_i \geq \widetilde{h}_c \right\} \right) \leq \sum_{i \neq c} \Pr_m \left(\widetilde{h}_i \geq \widetilde{h}_c \right), \tag{13}$$

where the last inequality follows from the union bound.

Statistics of the Dropout-Perturbed Logit Gap. For each competing class $i \neq c$, we define the perturbed logit gap between the deterministic top-1 class c and class i as

$$\widetilde{\Delta}_i = \widetilde{h}_c - \widetilde{h}_i. \tag{14}$$

This quantity directly measures whether class c remains ahead of class i after dropout perturbation. In particular, $\widetilde{\Delta}_i > 0$ means that class c still has a larger perturbed logit than class i, while $\widetilde{\Delta}_i \leq 0$ indicates that class i catches up with or surpasses class c.

Substituting the linear classifier $\widetilde{h}_r = a_r \widetilde{z} + b_r$ and the inverted dropout feature $\widetilde{z} = \frac{m}{q} \odot z$, we obtain

$$\begin{aligned} \widetilde{\Delta}_i &= (a_c \widetilde{z} + b_c) - (a_i \widetilde{z} + b_i) = (a_c - a_i)\widetilde{z} + b_c - b_i \\ &= \sum_{k=1}^{d} (a_c - a_i)_k \frac{m_k}{q} z_k + b_c - b_i. \end{aligned} \tag{15}$$

Therefore, the dropout perturbation affects the logit gap through each feature dimension k, weighted by both the feature value z_k and the classifier difference $(a_c - a_i)_k$.

Since inverted dropout rescales the retained features by $1/q$, it preserves the feature expectation:

$$\mathbb{E}_m \left[\frac{m_k}{q} \right] = \frac{1}{q} \mathbb{E}_m[m_k] = \frac{1}{q} \cdot q = 1. \tag{16}$$

Thus, the expected perturbed gap is

$$\begin{aligned} \mathbb{E}_m[\widetilde{\Delta}_i] &= \mathbb{E}_m \left[\sum_{k=1}^{d} (a_c - a_i)_k \frac{m_k}{q} z_k + b_c - b_i \right] \\ &= \sum_{k=1}^{d} (a_c - a_i)_k z_k + b_c - b_i = (a_c - a_i)^\top z + b_c - b_i \\ &= h_c - h_i = \gamma_i. \end{aligned} \tag{17}$$

This shows that the deterministic margin γ_i is exactly the mean of the dropout-perturbed logit gap. In other words, inverted dropout does not shift the average decision margin; it only introduces random fluctuations around this margin.

Next, we compute the variance of the perturbed gap. Since $b_c - b_i$ is a constant, it does not contribute to the variance. Moreover, the dropout masks $\{m_k\}_{k=1}^d$ are independent across feature dimensions, so the variance of the sum is the sum of variances. For each dimension:

$$\operatorname{Var}\left(\frac{m_k}{q} \right) = \frac{1}{q^2} \operatorname{Var}(m_k) = \frac{1}{q^2} q(1-q) = \frac{1-q}{q}. \tag{18}$$

Combining the Eq. 18 and Eq. 15, we can derive the following variance:

$$\begin{aligned}\mathrm{Var}_m(\widetilde{\Delta}_i) &= \mathrm{Var}_m\left(\sum_{k=1}^{d}(a_c-a_i)_k\frac{m_k}{q}z_k\right) = \sum_{k=1}^{d}\left((a_c-a_i)_k z_k\right)^2\mathrm{Var}\left(\frac{m_k}{q}\right)\\ &= \frac{1-q}{q}\sum_{k=1}^{d}\left((a_c-a_i)_k z_k\right)^2 := \frac{1-q}{q}V_i,\end{aligned} \tag{19}$$

where

$$V_i = \sum_{k=1}^{d}\left[(a_i-a_c)_k\, z_k\right]^2. \tag{20}$$

The quantity V_i characterizes how sensitive the logit gap between class c and class i is to dropout noise. If the feature dimensions with large activation values z_k also have large classifier differences $(a_i-a_c)_k$, then dropping these dimensions can cause a large fluctuation in the gap, leading to a larger V_i. Conversely, a small V_i means that the logit gap is relatively insensitive to dropout perturbations. Hence, $\mathrm{Var}_m(\widetilde{\Delta}_i)$ is determined by two factors: the dropout noise level $\frac{1-q}{q}$ and the class-specific sensitivity V_i.

Gaussian Approximation and Tail Bound. From the previous derivation, the perturbed logit gap $\widetilde{\Delta}_i$ is a sum of independent dropout-induced random variables plus a constant bias term. Specifically, each feature dimension contributes a random perturbation through the Bernoulli mask m_k. When the feature dimension d is sufficiently large and no single dimension dominates the total variance, the Central Limit Assumption allows us to approximate this weighted sum by a Gaussian random variable.

Therefore, using the mean and variance derived above, we have

$$\widetilde{\Delta}_i \approx \mathcal{N}\left(\gamma_i, \frac{1-q}{q}V_i\right). \tag{21}$$

This approximation indicates that the dropout-perturbed gap fluctuates around the deterministic margin γ_i, with variance controlled by the dropout noise level $\frac{1-q}{q}$ and the sensitivity term V_i.

The event that class i overtakes class c is equivalent to the event that the perturbed gap becomes non-positive:

$$\Pr_m(\widetilde{h}_i \geq \widetilde{h}_c) = \Pr_m(\widetilde{\Delta}_i \leq 0). \tag{22}$$

Let $\sigma_i^2 = \frac{1-q}{q}V_i$. When $V_i > 0$, we standardize the Gaussian variable as $\frac{\widetilde{\Delta}_i-\gamma_i}{\sigma_i} \approx \mathcal{N}(0,1)$. Thus,

$$\Pr_m(\widetilde{\Delta}_i \leq 0) \approx \Pr\left(\frac{\widetilde{\Delta}_i-\gamma_i}{\sigma_i} \leq \frac{0-\gamma_i}{\sigma_i}\right) = \Phi\left(-\frac{\gamma_i}{\sqrt{\frac{1-q}{q}V_i}}\right), \tag{23}$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution.

Since c is the deterministic top-1 class, we have $\gamma_i = h_c - h_i > 0$ for each $i \neq c$. Therefore, the standardized quantity

$$x_i = \frac{\gamma_i}{\sqrt{\frac{1-q}{q}V_i}} \tag{24}$$

is positive. Applying the standard Gaussian tail bound

$$\Phi(-x) \leq \exp\left(-\frac{x^2}{2}\right), \quad x > 0, \tag{25}$$

we obtain

$$\Pr_m(\widetilde{h}_i \geq \widetilde{h}_c) = \Pr_m(\widetilde{\Delta}_i \leq 0) \lesssim \exp\left(-\frac{1}{2}\cdot\frac{\gamma_i^2}{\frac{1-q}{q}V_i}\right) = \exp\left(-\frac{q}{2(1-q)}\cdot\frac{\gamma_i^2}{V_i}\right). \tag{26}$$

Method	Reference	Scheduler	Label Shift					Batch Size = 1					Mix Shift
			Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg	Avg
TENT	ICLR'21	Uniform	43.73	51.15	45.28	61.71	50.92	39.26	47.76	49.03	58.53	49.27	34.52
		Greedy	43.88	50.88	45.49	61.84	50.97	38.26	46.53	48.99	58.00	48.59	27.01
		Ours	**56.63**	**55.10**	**67.05**	**64.51**	**61.10**	**54.97**	**53.19**	**64.40**	**61.58**	**58.77**	**57.31**
SAR	ICLR'23	Uniform	45.87	52.63	56.98	62.92	55.18	29.74	43.55	57.12	56.68	47.91	56.05
		Greedy	45.66	52.42	58.70	62.40	55.40	28.75	42.58	55.79	55.70	46.84	55.26
		Ours	**58.79**	**57.66**	**69.26**	**66.63**	**63.37**	**54.73**	**53.95**	**64.17**	**60.58**	**58.60**	**60.29**
ActMAD	CVPR'23	Uniform	0.10	1.27	0.43	0.87	0.71	0.10	0.10	0.10	0.10	0.10	2.81
		Greedy	0.06	1.00	0.46	1.09	0.69	0.12	0.17	0.22	0.19	0.18	2.30
		Ours	**0.14**	**6.67**	**0.98**	**17.06**	**6.62**	**6.49**	**7.53**	**15.34**	**7.93**	**9.51**	**29.54**
CMF	ICLR'24	Uniform	71.93	74.62	82.09	79.72	77.43	17.15	30.97	40.68	45.06	34.55	60.29
		Greedy	68.20	73.00	81.92	77.62	75.65	16.36	30.26	39.65	44.13	33.68	59.59
		Ours	**73.19**	**75.52**	**82.99**	**80.56**	**78.39**	**51.12**	**44.39**	**60.60**	**54.72**	**52.81**	**61.23**
ReCAP	ICML'25	Uniform	53.19	58.77	70.36	68.79	63.42	44.54	53.54	66.07	65.02	58.14	59.35
		Greedy	52.64	58.06	69.66	68.04	62.73	44.02	52.47	65.35	64.34	57.38	58.61
		Ours	**60.38**	**59.50**	**72.26**	**68.85**	**65.57**	**62.17**	**61.62**	**72.82**	**70.26**	**67.02**	**61.41**
ZeroSiam	ICLR'26	Uniform	45.25	52.50	64.59	63.08	57.10	40.65	48.45	61.62	59.43	53.33	55.13
		Greedy	45.06	51.88	64.37	62.65	56.72	39.78	47.55	60.43	58.60	52.38	54.39
		Ours	**58.70**	**57.34**	**69.32**	**66.51**	**63.25**	**57.10**	**55.14**	**66.83**	**63.89**	**60.98**	**59.58**

Table 6: Experimental results for ViT-Base on ImageNet-C (severity level 5) under three wild settings with the **Sliding-Window** budget ratio $r = 0.5$. We report accuracy (%) averaged over three independent runs. The best and second-best results among the three schedulers in each comparison are highlighted in **bold** and underlined, respectively.

This inequality gives an interpretable upper bound for the probability that a single competing class i surpasses the deterministic top-1 class c under dropout. The probability decreases exponentially with the squared margin γ_i^2, meaning that a larger deterministic confidence gap makes the prediction more stable. In contrast, the probability increases when V_i is large, since a larger V_i implies stronger dropout-induced fluctuation of the corresponding logit gap.

Finally, combining this pairwise bound with the union bound over all competing classes $i \neq c$, we obtain

$$P_{\text{top1-dis}} \leq \sum_{i \neq c} \Pr_m(\widetilde{h}_i \geq \widetilde{h}_c) \lesssim \sum_{i \neq c} \exp\left(-\frac{q}{2(1-q)} \cdot \frac{\gamma_i^2}{V_i}\right). \tag{27}$$

This shows that the dropout-induced top-1 disagreement is controlled by the ratio between the squared deterministic margin γ_i^2 and the dropout sensitivity V_i. A larger margin makes the prediction more stable, while a larger V_i indicates that the corresponding logit gap is more sensitive to dropout perturbations. Hence, V_i serves as the dropout-induced sensitivity of the logit gap between class i and the deterministic top-1 class c. □

B Further Experiments

In this section, we provide further empirical evidence to validate the superiority and practicality of ECO-TTA. We first report complete results under various budget settings. We then compare ECO-TTA with the greedy strategy, highlighting its advantages in both performance and budget control. Furthermore, we evaluate ECO-TTA with six representative TTA methods under multiple budget settings using a ResNet backbone, demonstrating its generalizability across different model architectures. Together, these analyses demonstrate the potential of ECO-TTA as a lightweight and autonomous solution for real-world edge deployment.

B.1 Evaluation on Various Budget Settings

Together with the main-paper results, we report additional evaluation results in Tables 6–9 , covering budget ratios $r \in \{0.1, 0.2, 0.5\}$ under both the *global* and *sliding-window* protocols. These experiments include six base TTA methods and three wild TTA scenarios, all following the same experimental protocols described in the main paper. The results lead to the following observations, further supporting and complementing the analysis in the main text.

Method	Reference	Scheduler	Label Shift					Batch Size = 1					Mix Shift
			Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg	Avg
TENT	ICLR'21	Uniform	43.73	51.15	45.28	61.71	50.92	39.26	47.76	49.03	58.53	49.27	34.52
		Greedy	46.83	53.70	41.02	60.42	50.74	43.21	50.59	43.58	61.04	50.03	14.92
		Ours	**57.33**	**56.34**	**67.45**	**65.22**	**61.87**	**56.82**	**55.83**	**65.87**	**64.26**	**60.95**	**58.59**
SAR	ICLR'23	Uniform	45.87	52.63	56.98	62.92	55.18	29.74	43.55	57.12	56.68	47.91	56.05
		Greedy	49.00	55.28	48.23	65.19	54.79	32.72	45.31	58.71	57.52	49.62	57.23
		Ours	**59.54**	**58.56**	**69.88**	**67.59**	**64.18**	**58.62**	**57.30**	**67.76**	**64.79**	**62.35**	**61.10**
ActMAD	CVPR'23	Uniform	0.10	1.27	0.43	0.87	0.71	0.10	0.10	0.10	0.10	0.10	2.81
		Greedy	0.10	0.51	0.30	0.10	0.26	0.10	0.10	0.10	0.10	0.10	1.17
		Ours	**7.83**	**6.53**	**11.46**	**9.98**	**9.02**	**10.95**	**5.96**	**16.09**	**11.94**	**11.25**	**29.60**
CMF	ICLR'24	Uniform	71.93	74.62	82.09	79.72	77.43	17.15	30.97	40.68	45.06	34.55	60.29
		Greedy	70.07	72.32	80.12	79.26	75.80	17.29	31.08	40.66	45.24	34.65	60.11
		Ours	**74.98**	**75.41**	**82.98**	**80.70**	**78.75**	**51.66**	**44.82**	**60.33**	**55.47**	**53.16**	**61.42**
ReCAP	ICML'25	Uniform	53.19	58.77	70.36	68.79	63.42	44.54	53.54	66.07	65.02	58.14	59.35
		Greedy	54.65	51.84	71.05	69.58	62.26	43.18	55.49	67.42	63.06	58.23	58.03
		Ours	**60.70**	**60.56**	**72.27**	**69.96**	**66.22**	**61.86**	**62.28**	**72.81**	**71.02**	**67.33**	**62.03**
ZeroSiam	ICLR'26	Uniform	45.25	52.50	64.59	63.08	57.10	40.65	48.45	61.62	59.43	53.33	55.13
		Greedy	48.45	54.97	66.94	65.36	59.63	40.11	47.46	59.98	58.17	52.18	54.22
		Ours	**59.59**	**58.66**	**70.00**	**67.72**	**64.29**	**58.91**	**57.26**	**68.99**	**66.76**	**63.25**	**60.69**

Table 7: Experimental results for ViT-Base on ImageNet-C (severity level 5) under three wild settings with the **Global** budget ratio $r = 0.5$. We report accuracy (%) averaged over three independent runs.

Method	Reference	Scheduler	Label Shift					Batch Size = 1					Mix Shift
			Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg	Avg
TENT	ICLR'21	Uniform	30.12	40.47	52.74	53.38	45.11	26.85	37.72	50.58	50.88	42.48	49.68
		Greedy	31.09	40.36	53.49	53.28	45.45	25.43	36.87	49.45	50.07	41.46	49.07
		Ours	**52.62**	**48.82**	**62.27**	**58.75**	**55.81**	**51.25**	**46.74**	**60.38**	**56.59**	**53.91**	**55.74**
SAR	ICLR'23	Uniform	32.18	42.01	54.40	53.76	46.48	18.25	33.14	42.90	47.59	36.62	50.37
		Greedy	33.25	42.24	54.30	53.64	46.70	17.66	32.12	42.52	46.78	35.91	49.62
		Ours	**54.78**	**51.02**	**64.27**	**60.96**	**57.96**	**52.87**	**47.83**	**61.99**	**57.23**	**55.12**	**56.89**
ActMAD	CVPR'23	Uniform	0.12	2.02	0.57	2.38	1.35	0.28	0.62	0.74	0.72	0.61	1.32
		Greedy	0.16	4.22	1.97	4.39	2.85	0.19	0.44	0.80	0.60	0.53	10.80
		Ours	**10.12**	**8.44**	**5.49**	**10.90**	**8.65**	**9.25**	**7.60**	**14.56**	**15.00**	**11.76**	**29.05**
CMF	ICLR'24	Uniform	50.06	57.28	69.93	68.14	62.11	16.62	30.42	39.33	44.10	33.68	58.63
		Greedy	50.36	55.63	68.90	63.40	60.19	15.54	29.60	38.10	43.38	32.73	57.68
		Ours	**59.04**	**57.77**	**70.65**	**69.35**	**64.55**	**50.75**	**43.72**	**60.25**	**54.23**	**52.34**	**60.37**
ReCAP	ICML'25	Uniform	43.79	50.70	63.57	62.38	55.86	27.41	41.50	55.45	54.89	45.97	56.20
		Greedy	44.43	49.84	63.46	61.57	55.52	25.91	40.49	54.50	54.03	44.92	55.94
		Ours	**57.30**	**51.55**	**68.13**	**63.29**	**60.25**	**56.68**	**53.99**	**66.73**	**63.22**	**60.39**	**59.04**
ZeroSiam	ICLR'26	Uniform	31.37	41.22	54.41	54.39	46.28	27.04	38.03	50.73	51.19	42.73	50.46
		Greedy	32.17	41.18	54.63	53.96	46.37	25.83	37.11	50.11	50.25	41.82	49.55
		Ours	**54.16**	**50.76**	**64.37**	**60.82**	**57.75**	**53.24**	**48.59**	**62.58**	**58.49**	**55.89**	**57.87**

Table 8: Experimental results for ViT-Base on ImageNet-C (severity level 5) under three wild settings with the **Sliding-Window** budget ratio $r = 0.1$. We report accuracy (%) averaged over three independent runs.

1) Consistent and Superior Gains across Budget Ratios. Across the four supplementary configurations, comprising 72 combinations of base TTA methods and wild TTA scenarios, ECO-TTA achieves an average improvement of 9.86% over uniform allocation and consistently outperforms the baseline in every setting. These results are consistent with the findings in the main paper and demonstrate that the effectiveness of ECO-TTA is not tied to a specific budget ratio.

2) Independence from Long-Horizon Budget Accumulation. Across the three budget ratios, ECO-TTA achieves average improvements of 10.85% and 9.40% under the global and sliding-window protocols, respectively. While long-horizon global scheduling allows the scheduler to fully exploit its allocation flexibility, ECO-TTA still consistently outperforms rule-based uniform scheduling when budget decisions are restricted to local windows. This suggests that its benefit does not rely solely on long-horizon budget accumulation, but also comes from effective online budget allocation.

Method	Reference	Scheduler	Label Shift					Batch Size = 1					Mix Shift
			Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg	Avg
TENT	ICLR'21	Uniform	36.74	45.44	54.47	57.25	49.26	31.54	41.44	53.54	53.97	46.03	52.53
		Ours	**54.68**	**53.34**	**64.49**	**61.99**	**58.89**	**53.84**	**52.15**	**62.57**	**59.20**	**57.15**	**57.03**
SAR	ICLR'23	Uniform	39.35	46.63	58.40	57.55	51.22	21.07	36.63	48.81	51.15	40.64	53.42
		Ours	**57.69**	**56.79**	**67.80**	**65.63**	**62.26**	**56.63**	**54.68**	**65.30**	**60.72**	**59.51**	**60.40**
ActMAD	CVPR'23	Uniform	0.25	2.11	0.96	3.21	1.72	0.17	0.13	0.67	0.37	0.35	6.19
		Ours	**10.19**	**6.87**	**5.96**	**17.47**	**10.12**	**7.05**	**5.92**	**16.03**	**11.93**	**10.44**	**29.44**
CMF	ICLR'24	Uniform	58.26	62.01	71.51	68.32	65.48	16.68	30.63	39.76	44.63	34.01	59.62
		Ours	**65.04**	**65.07**	**75.69**	**71.30**	**69.56**	**51.39**	**45.17**	**61.10**	**55.93**	**53.53**	**60.91**
ReCAP	ICML'25	Uniform	48.50	54.77	67.36	65.47	59.73	34.31	46.35	60.27	59.13	51.06	57.87
		Ours	**59.44**	**54.99**	**71.36**	**67.84**	**63.67**	**60.97**	**60.46**	**71.63**	**69.49**	**65.95**	**61.34**
ZeroSiam	ICLR'26	Uniform	38.43	46.69	59.44	58.03	51.46	32.63	42.05	55.75	54.87	47.24	53.03
		Ours	**57.64**	**56.35**	**68.25**	**65.34**	**62.18**	**57.13**	**55.03**	**66.74**	**64.67**	**61.14**	**60.86**

Table 9: Experimental results for ViT-Base on ImageNet-C (severity level 5) under three wild settings with the **Global** budget ratio $r = 0.2$. We report accuracy (%) averaged over three independent runs.

Method	Reference	Scheduler	Label Shift					Batch Size = 1					Mix Shift
			Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg	Avg
TENT	ICLR'21	Uniform	28.75	15.98	32.33	37.81	28.72	1.30	0.98	12.22	7.72	5.84	22.57
		Ours	**30.09**	**19.15**	**39.47**	**41.29**	**32.66**	**23.80**	**16.28**	**42.49**	**40.60**	**31.26**	**26.64**
SAR	ICLR'23	Uniform	29.42	20.98	44.26	38.31	33.50	26.58	20.46	47.62	38.28	33.68	36.72
		Ours	**34.02**	**24.05**	**45.69**	**41.68**	**36.52**	**37.31**	**27.26**	**49.20**	**45.04**	**39.86**	**37.85**
ActMAD	CVPR'23	Uniform	6.00	2.75	12.74	8.42	7.58	0.95	0.69	2.56	1.97	1.58	13.51
		Ours	**11.30**	**6.21**	**38.75**	**27.52**	**21.59**	**9.69**	**6.16**	**33.19**	**18.43**	**17.35**	**14.41**
CMF	ICLR'24	Uniform	21.81	18.93	32.37	28.90	25.75	20.19	16.04	42.84	29.33	27.56	40.57
		Ours	**22.65**	**19.31**	**33.90**	**30.23**	**26.78**	**21.22**	**18.17**	**44.69**	**33.09**	**29.83**	**41.83**
ReCAP	ICML'25	Uniform	42.03	18.09	43.41	44.11	36.57	39.82	28.91	56.15	50.52	44.12	40.75
		Ours	**42.88**	**21.74**	**56.71**	**47.11**	**42.06**	**46.50**	**33.79**	**60.93**	**57.28**	**49.83**	**41.95**
ZeroSiam	ICLR'26	Uniform	31.03	22.52	49.39	41.33	36.40	38.75	28.38	53.64	47.35	42.25	37.74
		Ours	**34.69**	**25.87**	**51.67**	**45.59**	**39.77**	**42.76**	**32.86**	**58.01**	**52.20**	**46.70**	**41.81**

Table 10: Experimental results for ResNet-50 on ImageNet-C (severity level 5) under three wild settings with the **Global** budget ratio $r = 0.5$. We report accuracy (%) averaged over three independent runs.

3) Robustness to Extreme Data Streams and Negative Adaptation. Across all evaluated budget-allocation settings, ECO-TTA achieves substantial improvements in the batch-size-one scenario. It also consistently mitigates severe performance degradation or model collapse for unstable adaptation methods such as ActMAD and TENT. These results further corroborate the robustness of ECO-TTA to low-data test streams and negative adaptation, as observed in the main paper.

B.2 Comparison with Greedy Scheduler

We further supplement the results in Tables 6–8 with the *Greedy* scheduling baseline, which is omitted from the main paper. This baseline spends the available budget as early as possible. The results further support the observations in the main paper and demonstrate the advantage of ECO-TTA in budget-aware scheduling.

1) Spending the Budget Early Does Not Yield Consistent Gains. We evaluate both uniform and greedy scheduling across 54 experimental configurations. Greedy scheduling performs worse than uniform scheduling in 39 settings, achieves only marginal improvements in 14 settings, and produces one tie, leading to an overall performance drop of 0.72%. This finding supports our reformulation of budget-constrained TTA as an online investment problem: simple rule-based schedulers, such as greedy or uniform allocation, are insufficient to fully exploit the potential value of adaptation.

2) Myopic Budget Consumption Misses More Valuable Future Adaptation Opportunities. The limitation of greedy scheduling is particularly evident under the Mix Shift scenario, where the test stream contains samples from multiple shifted domains. Under the global and sliding-window protocols with $r = 0.5$, greedy scheduling performs, on average, 3.75% and 1.83% worse than uniform

scheduling on Mix Shift, respectively. In wild test streams, the distribution shift encountered by different batches may change rapidly. Concentrating the budget on early batches may therefore waste substantial resources on a specific shift and leave insufficient computation for newly emerging shifts. This further supports the central insight of the main paper: under limited budgets, a TTA scheduler must determine *when adaptation is worth its cost*.

3) ECO-TTA Consistently Outperforms Greedy Scheduling across Methods and Settings. Across all evaluated settings, ECO-TTA consistently surpasses the greedy scheduler, achieving an average improvement of 10.47%. By estimating the value of each incoming batch and applying an online Lagrangian decision rule, ECO-TTA allocates the limited budget to adaptation operations whose expected gains justify their costs. Consequently, it consistently outperforms rule-based scheduling strategies.

B.3 Evaluation on ResNet

To further validate the generalizability of ECO-TTA across different backbone architectures, we evaluate our method using a ResNet-50 backbone. We consider the same six representative TTA methods and three budget ratios as in the main paper. Since ECO-TTA is implemented independently of the backbone architecture, all ResNet experiments follow the same hyperparameter settings and implementation details used in the main paper. Tables 10–12 report the results under different budget ratios. ECO-TTA outperforms the baseline in all 54 experimental configurations, achieving an average improvement of 4.90%. Consistent with the observations on the ViT-Base backbone, ECO-TTA brings particularly substantial gains in the batch-size-one setting, with an average improvement of 8.45%. These consistent improvements on both ViT-Base and ResNet-50 demonstrate that ECO-TTA is a lightweight and model-agnostic scheduling module, rather than a method that relies on the architectural characteristics of a specific backbone.

Method	Reference	Scheduler	Label Shift					Batch Size = 1					Mix Shift
			Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg	Avg
TENT	ICLR'21	Uniform	25.03	18.60	41.28	36.84	30.80	2.12	7.15	19.89	32.08	16.19	31.34
		Ours	**25.36**	**19.35**	**43.87**	**39.83**	**32.55**	**7.56**	**17.60**	**46.26**	**37.65**	**28.58**	**33.90**
SAR	ICLR'23	Uniform	23.71	19.46	42.33	34.57	30.44	23.48	19.21	47.99	33.75	31.62	33.52
		Ours	**26.79**	**21.00**	**44.15**	**38.22**	**32.92**	**30.07**	**22.06**	**48.61**	**41.65**	**35.97**	**35.76**
ActMAD	CVPR'23	Uniform	12.64	6.13	27.68	18.72	16.54	3.70	2.54	10.41	7.57	6.21	17.03
		Ours	**16.28**	**7.33**	**36.74**	**27.18**	**22.26**	**6.61**	**8.67**	**32.53**	**18.69**	**17.29**	**18.37**
CMF	ICLR'24	Uniform	33.31	25.31	51.04	44.34	38.85	21.92	18.22	45.04	32.24	29.85	38.42
		Ours	**33.76**	**27.23**	**53.90**	**45.53**	**40.53**	**22.34**	**19.37**	**46.95**	**34.65**	**31.39**	**41.42**
ReCAP	ICML'25	Uniform	32.63	21.65	41.48	40.41	34.14	28.91	21.41	47.64	40.15	34.90	37.46
		Ours	**36.19**	**23.57**	**42.61**	**46.52**	**37.29**	**42.15**	**31.73**	**56.03**	**46.63**	**44.27**	**39.96**
ZeroSiam	ICLR'26	Uniform	24.41	19.14	47.54	35.27	32.07	30.10	21.47	48.54	40.60	35.52	35.05
		Ours	**26.32**	**20.81**	**48.07**	**40.49**	**34.43**	**34.45**	**24.78**	**51.25**	**45.60**	**39.32**	**37.02**

Table 11: Experimental results for ResNet-50 on ImageNet-C (severity level 5) under three wild settings with the **Global** budget ratio $r = 0.1$. We report accuracy (%) averaged over three independent runs.

C Additional Ablation Study

In the main paper, we provide a comprehensive analysis of the effects of two key hyperparameters: the memory capacity K and the saturation rate of additional steps β. In this section, we extend the analysis by further investigating the sensitivity of two other critical hyperparameters: the shadow-price update rate η and the dropout keep probability q. And we further provide the visualization of the allocation process under different schedules. Consistent with the main paper, all experiments are conducted under the ImageNet-C (level-5) label-shift wild setting, using ReCAP as the base TTA method. These results provide additional insights into the effectiveness and robustness of our method.

C.1 Sensitivity of η in ECO-TTA

The shadow-price update rate η controls how quickly ECO-TTA adjusts the estimated opportunity cost of adaptation actions under the given budget constraint. As shown in Fig. 5(a), increasing η from

Method	Reference	Scheduler	Label Shift					Batch Size = 1					Mix Shift
			Noise	Blur	Wea.	Digi.	Avg	Noise	Blur	Wea.	Digi.	Avg	Avg
TENT	ICLR'21	Uniform	25.48	18.44	38.07	37.64	30.20	4.26	2.95	18.00	20.80	11.99	29.07
		Ours	**27.35**	**19.74**	**41.80**	**40.62**	**32.71**	**15.62**	**16.33**	**39.71**	**35.67**	**27.58**	**30.39**
SAR	ICLR'23	Uniform	25.55	19.50	43.08	36.27	31.47	23.95	19.52	47.01	34.95	31.85	34.40
		Ours	**30.01**	**21.87**	**44.08**	**39.51**	**34.12**	**33.90**	**24.32**	**47.51**	**43.30**	**37.48**	**36.72**
ActMAD	CVPR'23	Uniform	8.91	4.59	19.66	13.52	11.85	2.29	1.29	6.12	4.57	3.65	16.09
		Ours	**14.54**	**10.21**	**37.97**	**26.77**	**22.89**	**7.51**	**7.83**	**34.50**	**15.76**	**16.99**	**20.09**
CMF	ICLR'24	Uniform	32.06	25.16	48.35	42.14	37.25	20.19	17.79	46.06	31.40	29.44	39.45
		Ours	**32.69**	**26.00**	**48.73**	**42.72**	**37.86**	**22.13**	**19.13**	**46.24**	**34.24**	**30.99**	**41.72**
ReCAP	ICML'25	Uniform	36.27	21.18	51.91	41.06	37.69	33.12	23.76	51.07	44.45	38.43	37.64
		Ours	**38.73**	**23.33**	**52.38**	**45.88**	**40.17**	**44.19**	**34.07**	**59.11**	**53.58**	**47.97**	**40.95**
ZeroSiam	ICLR'26	Uniform	26.34	19.88	47.91	37.69	33.40	33.60	23.95	50.04	43.43	38.03	34.27
		Ours	**29.98**	**22.30**	**48.82**	**42.63**	**36.33**	**37.71**	**28.02**	**53.94**	**48.04**	**42.21**	**37.29**

Table 12: Experimental results for ResNet-50 on ImageNet-C (severity level 5) under three wild settings with the **Global** budget ratio $r = 0.2$. We report accuracy (%) averaged over three independent runs.

a very small value consistently improves the accuracy under both budget ratios, i.e., $r = 0.1$ and $r = 0.5$. This indicates that a proper update rate enables the scheduler to better respond to the current adaptation demand and allocate optimization steps to more valuable samples. The performance reaches its best around 0.5 on the plotted scale, corresponding to $\eta = 5 \times 10^{-3}$. Based on these results, we set this as the default value. However, when η becomes too large, the accuracy gradually decreases. This is because an overly aggressive update makes the shadow price fluctuate too strongly, causing unstable budget allocation and suppressing useful adaptation opportunities. Overall, ECO-TTA remains consistently above the baseline across a broad range of η, demonstrating that the proposed scheduling mechanism is robust to the choice of the shadow-price update rate.

C.2 Sensitivity of q in ECO-TTA

The dropout keep probability q determines the perturbation strength used to estimate prediction disagreement. As illustrated in Fig. 5(b), the performance first improves as q increases and achieves the best accuracy at around $q = 0.5$ for both $r = 0.1$ and $r = 0.5$. This suggests that moderate dropout perturbations provide an informative disagreement signal, allowing ECO-TTA to identify samples that benefit more from additional adaptation. When q is too small, excessive feature dropping introduces overly noisy perturbations, making the disagreement proxy less reliable. In contrast, when q is too large, the perturbation becomes too weak to reveal meaningful prediction instability. In the extreme case of $q = 1$, dropout is effectively disabled, and the disagreement signal degenerates, leading to a clear performance drop. The results show that ECO-TTA performs robustly within the range $[0.1, 0.9]$ and consistently outperforms the baseline in this interval. Therefore, we choose $q = 0.5$ as the default setting.

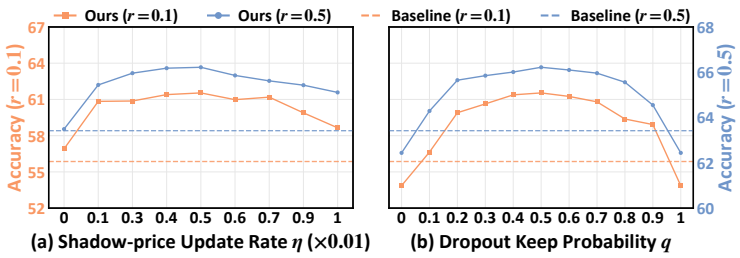


Figure 5: Performance with (a) different shadow-price update rate η (b) varying dropout keep probability q in proxy.

D More Implementation Details

D.1 Base TTA Methods

We validate ECO-TTA on the following representative base TTA methods. TENT (Wang et al., 2020) adapts the model by minimizing the prediction entropy of incoming test samples, updating normalization parameters to produce more confident predictions. ActMAD (Mirza et al., 2023) aligns the activation statistics of test samples with source-domain statistics across multiple network layers, providing fine-grained feature-level supervision for adaptation. SAR (Niu et al., 2023) combines entropy-based sample filtering with sharpness-aware optimization, steering adaptation toward flatter minima and improving stability under challenging wild test streams. CMF (Lee & Chang, 2024) adopts a teacher-student framework, where the teacher model provides stable pseudo-label supervision and is continually updated using Kalman filtering to balance adaptation to distribution shifts with resistance to noisy model updates. ReCAP (Hu et al., 2025) replaces conventional sample-wise entropy with an efficient region-confidence proxy derived from probabilistic local-region modeling, encouraging confident and locally consistent predictions. ZeroSiam (Chen et al., 2026a) introduces a lightweight asymmetric Siamese structure with a learnable predictor and a stop-gradient operation to prevent prediction collapse and regularize biased adaptation signals. For all these methods, ECO-TTA preserves their original adaptation objectives, optimizers, trainable parameters, hyperparameters, and sample-selection rules, modifying only the timing of adaptation.

D.2 Additional Details of Historical Memory

Memory Mechanism. The historical memory is designed to provide ECO-TTA with empirical evidence about the relationship between test-batch statistics and adaptation payoff. Since the expected adaptation value estimator relies on previously observed outcomes, the memory must contain a small number of reliable initial records before value-based scheduling can be activated. To this end, we introduce a brief warm-up stage at the beginning of the test stream. Specifically, for the first three test batches, ECO-TTA performs the one-step adaptation action $a^{(1)}$ and records the resulting adaptation gains together with the corresponding feature statistics in $\mathcal{M}$. These initial records serve as seed decision assets, allowing the estimator to compare incoming batches with past adaptation cases and estimate their expected value.

After the warm-up stage, ECO-TTA updates the memory online by inserting newly observed adaptation outcomes. The memory bank grows until it reaches its maximum capacity of $K = 50$. Once the capacity is reached, $\mathcal{M}$ is updated in a first-in-first-out manner, so that recent adaptation experiences are retained while outdated records are gradually removed. This design naturally matches the online TTA setting, where the target distribution may evolve over time and recent observations are often more informative for future scheduling decisions. Since the warm-up stage requires only three one-step adaptations and the memory stores at most 50 batch-level records, the storage and computation overhead remains small, making the mechanism practical for budget-aware online adaptation.

Reset Indicator. Following the model-recovery principle of common practice (Niu et al., 2022; 2023; Hu et al., 2025), we use prediction entropy as a lightweight indicator of potential model collapse. SAR observes that a collapsed model may produce abnormally overconfident predictions, leading to persistently low prediction entropy. Based on this observation, our reset mechanism detects potential collapse by monitoring whether the current prediction entropy drops abnormally relative to the model's recent prediction behavior. Specifically, let C_t denote the mean prediction entropy of the current batch, and let $\bar{C}_{t-1}$ denote the exponential moving average of the mean prediction entropy over the preceding batches. The reset indicator is activated when $C_t < \bar{C}_{t-1} - \delta$,, and the reset action $a^{(\text{reset})}$ is then executed at the decision step. Here, δ prevents ordinary batch-level entropy fluctuations from triggering unnecessary recovery and is set to 5×10^{-3} by default.

For base TTA methods without an inherent reset mechanism, such as TENT, we use the above entropy-based indicator as a unified recovery strategy. For methods with built-in reset mechanisms, such as SAR and ReCAP, we retain their original collapse indicators and recovery operations. This reset design is highly efficient, requiring only entropy statistics and an exponential moving average. We measure the wall-clock cost of the reset action $a^{(\text{reset})}$ relative to a single update step $a^{(1)}$, and set the normalized reset cost C_{reset} according to the observed runtime measurements. Specifically,

C_{reset} is set to 0.4 for our entropy-based reset, and to 0.7 and 0.5 for the built-in reset mechanisms of SAR and ReCAP, respectively.

D.3 Additional Details of Uniform and Greedy Schedulers

Rule-Based Scheduling Baselines. In the experiments and ablation studies of the main paper, we use *Uniform* and *Greedy* as two limited-budget scheduling baselines. *Uniform* distributes the adaptation budget evenly across the entire test stream. For example, when the budget ratio is 0.5, Uniform performs adaptation on one out of every two consecutive batches and skips the other. In contrast, *Greedy* consumes the available budget as early as possible. It adapts on every incoming batch until the budget is exhausted, after which no further adaptation is performed. For example, given a budget ratio of 0.5 and a test stream of 1,000 batches, Greedy adapts on the first 500 batches and skips the remaining 500 batches.

Allocation Pattern Visualization. We provide a direct visualization comparing the allocation processes of different scheduling strategies. As shown in Fig. 6, Uniform and Greedy exhibit fixed rule-based allocation patterns that are independent of the actual adaptation value of incoming batches. As a result, both baselines may inevitably spend valuable adaptation budget on unhelpful or even harmful updates. In contrast, ECO-TTA selectively allocates the budget across the test stream according to the estimated adaptation value, enabling it to prioritize batches where adaptation is more likely to be beneficial.

Behavior under Sliding-Window Budgets. Since Uniform distributes the budget evenly, it is unaffected by the additional locality constraint imposed by the sliding-window protocol and therefore behaves identically under the global and sliding-window protocols. Greedy, however, is constrained by the budget assigned to each sliding window. For example, with a window size of 100 and a budget ratio of 0.5, Greedy performs adaptation on the first 50 batches of every window and skips the remaining 50 batches. This local greedy behavior still follows the same principle of early budget consumption, but prevents the scheduler from exhausting the entire budget at the beginning of the full test stream.

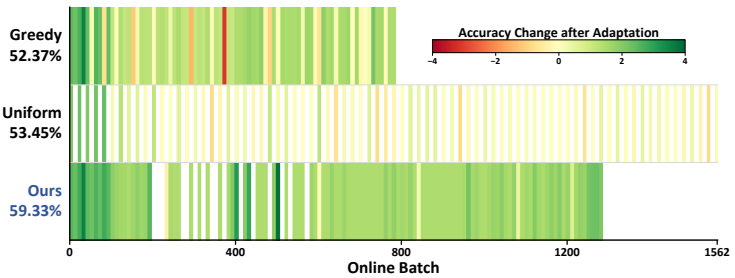


Figure 6: Visualization of budget allocation patterns produced by Uniform, Greedy, and ECO-TTA. ReCAP is used as the base TTA method and evaluated on the Gaussian-noise domain of the ImageNet-C label-shift stream.

D.4 Pseudocode of our ECO-TTA

Algorithm 1 outlines the complete online execution procedure of ECO-TTA. Operating at the batch level, ECO-TTA coordinates the Expected Adaptation Value Estimator and the Shadow-Price Budget Controller through a sequential decision process. The procedure consists of two main steps. First, ECO-TTA estimates the expected value of each candidate adaptation action using the feature statistics of the current batch and the adaptation outcomes stored in historical memory. Second, under the remaining budget constraint, it selects and executes the feasible action with the highest Lagrangian utility. After execution, ECO-TTA records the observed adaptation gain in historical memory and updates both the remaining budget and the shadow price to guide subsequent adaptation decisions.

Algorithm 1: ECOnomics-guided **TTA** framework (ECO-TTA).

Input: Test stream $\mathcal{X} = \{x_t\}_{t=1}^{T}$, source model θ_{src}, base TTA method $\mathcal{U}$, action space $\mathcal{A} = \{a^{(0)}, \ldots, a^{(S)}, a^{(\text{reset})}\}$, total budget B, memory capacity K, warm-up length T_{w}, hyperparameters β and η.
Output: Prediction sequence $\{\hat{y}_t\}_{t=1}^{T}$.

Initialize model $\theta_1 \leftarrow \theta_{\text{src}}$, memory $\mathcal{M} \leftarrow \emptyset$, remaining budget $b_1 \leftarrow B$, shadow price $\lambda_1 \leftarrow 0$, reset indicator $R_0 \leftarrow 0$;
Set the planned spending rate $\rho_B \leftarrow B/T$;
for $x_t \in \mathcal{X}$ **do**
  Obtain prediction $\hat{y}_t$ using the current model;
  Extract batch statistics (μ_t, σ_t^2);
  `// 1. Select an adaptation action`
  if $t \leq T_{\text{w}}$ **then**
    Select the one-step action $a_t \leftarrow a^{(1)}$;
  else
    Compute similarity weights $\{\omega_{ti}\}_{i \in \mathcal{M}}$ from batch statistics;
    Estimate the multi-step adaptation value $a^{(s)}$, $s \in \{0, 1, \ldots, S\}$; `// (Eq. 6)`
    Set the value of $a^{(\text{reset})}$ according to the indicator R_{t-1};
    Construct the feasible action set:

$$\mathcal{A}_t^{\text{feas}} \leftarrow \{a \in \mathcal{A} \mid c(a) \leq b_t\};$$

    Select the action with the largest Lagrangian utility ; `// (Eq. 9)`
  end
  `// 2. Execute the selected action`
  if $a_t = a^{(s)}$ *with* $s \geq 1$ **then**
    Apply the base method $\mathcal{U}$ for one update step;
    Compute proxy accuracy $\widehat{A}_{\text{after}}^{(1)}$ and $\hat{g}_t^{(1)} \leftarrow \widehat{A}_{\text{after}}^{(1)} - \widehat{A}_{\text{before}}$; ; `// (Eq. 5)`
    Insert the one-step adaptation outcome $\mathcal{E}_t = (\mu_t, \sigma_t^2, \hat{g}_t^{(1)})$ into $\mathcal{M}$;
    Remove the oldest asset if $|\mathcal{M}| > K$;
    `// Complete the remaining steps of the selected action`
    if $s > 1$ **then**
      Apply the base method $\mathcal{U}$ for the remaining $s - 1$ update steps;
    end
  else
    Execute a_t, i.e., skip adaptation or reset the model;
  end
  Update remaining budget b_{t+1}, and shadow price λ_{t+1} ; `// (Eq. 7)`
  Update the reset indicator R_t using the corresponding recovery criterion;
end

E More Experimental Details

E.1 More Details on Dataset

ImageNet-C Dataset. ImageNet-C (Hendrycks & Dietterich, 2019) is a widely used robustness benchmark constructed by applying common visual corruptions to the ImageNet validation set (Deng et al., 2009). It contains 15 corruption types, including Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transform, pixelation, and JPEG compression, as shown in Fig. 7. Each corruption is further divided into five severity levels, where higher levels indicate stronger visual perturbations and larger

distribution shifts. We use ImageNet-C (level=5) to evaluate the robustness of TTA methods under synthetic but diverse covariate shifts.

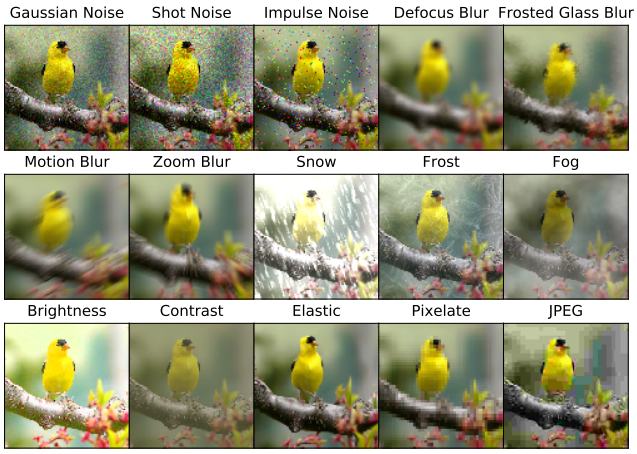


Figure 7: An illustration of representative corruption types in the ImageNet-C dataset.

ImageNet-R Dataset. ImageNet-R contains naturally occurring renditions of ImageNet categories, including paintings, cartoons, sculptures, sketches, toys, embroidery, origami, and other artistic styles (Hendrycks et al., 2021). Unlike ImageNet-C, which mainly introduces low-level corruptions, ImageNet-R focuses on style-level and appearance-level distribution shifts while preserving the semantic identity of objects, as shown in Fig. 8. This dataset provides a challenging benchmark for evaluating whether TTA methods can generalize beyond synthetic corruptions and adapt to real-world visual style variations.

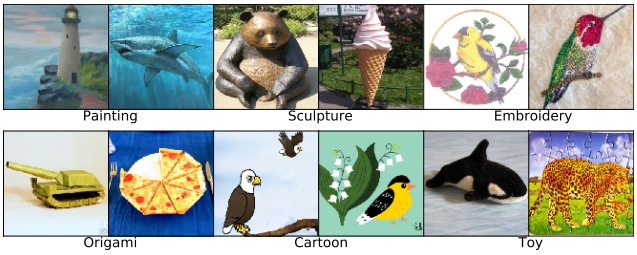


Figure 8: An illustration of representative rendition styles in the ImageNet-R dataset.

ImageNet-S Dataset. ImageNet-S (Wang et al., 2019), also known as ImageNet-Sketch, consists of sketch-style images corresponding to ImageNet categories. Compared with natural images, sketch images largely remove color and texture information while retaining object contours and high-level shape structures, as shown in Fig. 9. This creates a significant domain gap from the source ImageNet distribution. We use ImageNet-S to evaluate whether TTA methods can adapt under appearance-changing but structure-preserving distribution shifts, where robust prediction relies more heavily on shape-level cues.

E.2 More Details on TTA Scenarios

Mild TTA Setting. The mild TTA setting follows the standard online test-time adaptation protocol (Wang et al., 2020). During evaluation, the model receives unlabeled target samples sequentially and updates itself without accessing source data. The target stream is relatively stable, where samples are typically drawn from a fixed target distribution, such as one corruption type at a specific severity level in ImageNet-C, or one shifted dataset such as ImageNet-R or ImageNet-S. This setting mainly evaluates whether a method can effectively adapt to a consistent distribution shift while avoiding error accumulation during online updates.

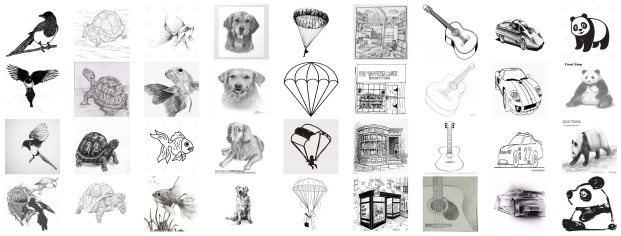


Figure 9: An illustration of representative sketch-style images in the ImageNet-S dataset.

Wild TTA Setting. The wild TTA setting is proposed by (Niu et al., 2023) to better reflect realistic and challenging deployment conditions. Unlike the mild setting, where the target stream is relatively stable, wild TTA introduces stronger uncertainty and non-stationarity into the online adaptation process. Following (Niu et al., 2023), we consider three representative wild settings: *Label Shift*, *Batch Size = 1*, and *Mix Shift*. In the *Label Shift* setting, the class distribution of the test stream is imbalanced and differs from the source distribution, making adaptation more difficult because pseudo-labels can be biased toward dominant classes. In the *Batch Size = 1* setting, the model receives only one test sample at each adaptation step, which removes reliable batch-level statistics and makes entropy-based or normalization-based adaptation less stable. In the *Mix Shift* setting, samples from different corruption types or domains are mixed within the test stream, resulting in a non-stationary target distribution whose shift pattern changes over time. Therefore, the model must handle label distribution changes, limited batch information, and dynamically varying covariate shifts while still operating without labels or source data. These settings place higher demands on the adaptation strategy, since unnecessary updates on unreliable samples may lead to negative transfer. They are particularly suitable for evaluating ECO-TTA, as our method aims to allocate limited adaptation budgets to more valuable samples while avoiding inefficient or harmful optimization.