FOZO: Forward-Only Zeroth-Order Prompt Optimization for Test-Time Adaptation

Supplementary Material

A. Implementation Details

A.1. General Experiment Setup

All experiments utilize the ViT-Base [4] model as the source model backbone, with its weights initialized from the timm repository [15]. For quantized models, we employ 8-bit PTQ4ViT [16]. All baseline methods utilize the hyperparameters reported in their respective papers.

A.2. Forward-Only Adaptation Methods

This section details the hyperparameters and specific configurations used for the back-free test-time adaptation methods evaluated.

FOZO adapts the model by learning new input prompts. We use 3 prompt embeddings, initialized uniformly. We employ an n-SPSA zero-order gradient estimator, where n is set to 1 as specified in Eqn. 6 for 2 forward passes, and set to 14 for 28 forward passes. The initial learning rate η in Eqn. 7 is 0.08. In Eqn. 12, The decay factor α is set to 0.9 , the threshold factor τ to 1.05, and the moving average factor β for the historical average loss to 0.9. The trade-off parameter λ in Eqn. 15 is set to 0.4. Source training statistics are estimated using the validation set of ImageNet-1K.

LAME [1]. We reproduce the LAME method, employing its reported hyperparameters. For Vision Transformer (ViT) features, the LAME method employs k-Nearest Neighbors (kNN) affinity, with the number of neighbors k set to 5.

T3A [7]. We reproduce the Test-Time Classifier Adjustment (T3A) method, adhering to its reported hyperparameters unless stated otherwise. The primary T3A hyperparameter, M, controls the number of low-entropy supports retained per class. While the original work explored values like $\{1, 5, 20, 50, 100, \text{N/A}\}$ (where N/A means retaining all samples), for our experiments, we consistently set M to 20.

FOA [12]. We reproduce the FOA method, adhering to their reported hyperparameters unless stated otherwise. FOA adapts the model by learning 3 new input prompts. Optimization is performed via a derivative-free CMA-ES algorithm with a population size of 28. The trade-off parameter λ of fitness function is set to 0.4 for ImageNet-C, V2, and Sketch, and 0.2 for ImageNet-R. For activation shifting, the step size γ is 1.0, and historical statistics are updated with an exponential moving average factor $\alpha = 0.1$.

ZOA [3]. We reproduce the ZOA method, adhering to their reported hyperparameters unless stated otherwise. For the ViT model, we add the perturbation vectors ϵ and $\nu$with the step size of 0.02 and 0.05, respectively. We set the learning rate ofθ and α to be 0.0005 and 0.01, respectively. We set the maximum number of domain knowledge parameters as N = 32. For multiple forward propagations, the steps is adjusted accordingly. For instance, 14 steps are employed for 28 forward passes.

A.3. Back-Based Adaptation Methods

This section details the hyperparameters and specific configurations used for the back-based test-time adaptation methods evaluated.

TENT [13]. We reproduce the TENT method, adhering to their reported hyperparameters unless stated otherwise. We employ SGD with 0.9 momentum, updating only the affine parameters of layer normalization layers. The standard batch size is 64. For the ViT model, the learning rate is 0.001.

DeYO [8]. We reproduce the DeYO method, adhering to their reported hyperparameters unless stated otherwise. We employ SGD with 0.9 momentum, updating only the affine parameters of layer normalization layers. For the ViT model, the learning rate is 0.00025. Key DeYO hyperparameters are $T_{\text{Ent}} = 0.4 \times \ln C$, $Ent_0 = 0.5 \times \ln C$, and T_{PLPD} is set to 0.2.

SAR [11]. We reproduce the SAR method, adhering to its reported hyperparameters. For the ViT model, we employ SGD with 0.9 momentum, updating only the affine parameters of Layer Normalization layers. Consistent with the original work, we freeze blocks9, blocks10, and blocks11 for adaptation. The learning rate is 0.001.

EATA [10]. We reproduce the EATA method, adhering to its reported hyperparameters unless stated otherwise. We employ SGD with 0.9 momentum and 0.0 weight decay, updating only the affine parameters of batch normalization layers. The learning rate is 0.00025, and the model is updated for 1 step per batch. Key EATA hyperparameters are

the entropy margin $E_0 = 0.4 \times \ln C$, the cosine similarity threshold $\epsilon = 0.05$, and the Fisher regularization trade-off parameter $\beta = 2000$. For calculating the Fisher information, 2000 source samples are utilized. The moving average factor α for updating model probabilities is set to 0.1.

A.4. Continual Adaptation

In this setup, the model undergoes continuous adaptation across all domains without re-initialization upon domain switching. This simulates a scenario where the model must continually adjust to sequential distribution shifts, building upon previous adaptations. We adhere to the RobustBench benchmark [2], using the ImageNet-C (5k) dataset, where only 5,000 images are sampled per domain (instead of 50,000). This choice simulates a resource-constrained online setting and focuses on the method's ability to adapt with limited data per shift.

A.5. Datasets

We used the following datasets to benchmark the performance of experiments:

ImageNet-C [5] is constructed by applying 15 distinct corruption types (e.g., Gaussian noise, motion blur, pixelation) across five severity levels to the original ImageNet validation set. This benchmark introduces diverse perturbations to simulate real-world image degradation, specifically designed to evaluate neural network robustness to common corruptions.

ImageNet-R [6] extends the ImageNet dataset to evaluate model generalization on non-natural or stylized images, such as cartoons, graffiti, embroidery, and sculptures. It comprises approximately 30,000 images across 200 ImageNet classes, all rendered in artistic or alternative mediums like paintings, origami, or animations.

ImageNet-Sketch [14] comprises hand-drawn sketches corresponding to all ImageNet images, designed to evaluate model performance on abstract, human-drawn artistic styles.

B. Proofs

In this section, we prove Theorems 1 and 2 proposed in the main paper. We begin by establishing the fundamental ℓ-smoothness descent lemma. Then, we decompose the SPSA gradient estimator using Taylor expansion and analyze its bias and variance by leveraging some fundamental assumptions previously introduced. These derived bounds are then substituted back into the descent lemma, ultimately leading to the convergence analysis of FOZO.

B.1. Basic Theorem: ℓ-Smoothness Descent Lemma

Based on the ℓ-smoothness assumption of the loss function $\mathcal{L}$, we have the standard quadratic upper bound:

$$\mathcal{L}(\mathbf{P}_{t+1}) \leq \mathcal{L}(\mathbf{P}_t) + \nabla\mathcal{L}(\mathbf{P}_t)^\top(\mathbf{P}_{t+1} - \mathbf{P}_t) + \frac{\ell}{2}\|\mathbf{P}_{t+1} - \mathbf{P}_t\|^2, \tag{B.1}$$

substituting the update rule $\mathbf{P}_{t+1} - \mathbf{P}_t = -\eta\widehat{\nabla}\mathcal{L}(\mathbf{P}_t; \mathcal{B}_t)$, we obtain:

$$\mathcal{L}(\mathbf{P}_{t+1}) \leq \mathcal{L}(\mathbf{P}_t) - \eta\nabla\mathcal{L}(\mathbf{P}_t)^\top\widehat{\nabla}\mathcal{L}(\mathbf{P}_t; \mathcal{B}_t) + \frac{\eta^2\ell}{2}\|\widehat{\nabla}\mathcal{L}(\mathbf{P}_t; \mathcal{B}_t)\|^2. \tag{B.2}$$

Taking the expectation on both sides conditioned on $\mathbf{P}_t$:

$$\mathbb{E}_t[\mathcal{L}(\mathbf{P}_{t+1})] - \mathcal{L}(\mathbf{P}_t) \leq -\eta\mathbb{E}_t[\nabla\mathcal{L}(\mathbf{P}_t)^\top\widehat{\nabla}\mathcal{L}(\mathbf{P}_t; \mathcal{B}_t)] + \frac{\eta^2\ell}{2}\mathbb{E}_t[\|\widehat{\nabla}\mathcal{L}(\mathbf{P}_t; \mathcal{B}_t)\|^2], \tag{B.3}$$

where the expectation $\mathbb{E}_t[\cdot]$ is taken over all randomness introduced at step t, including the data batch $\mathcal{B}_t \sim \mathcal{D}_{test}$ and the SPSA random perturbation $\mathbf{Z}_t \sim \mathcal{N}(0, I)$. Our task is to bound the two expectation terms on the right-hand side separately.

B.2. Decomposing the Gradient Estimator

To bound the expectation terms derived from the Eqn.B.3, we first analyze the SPSA gradient estimator. This section details its decomposition using Taylor expansion.

The SPSA gradient estimator at time t is defined as:

$$\widehat{\nabla}\mathcal{L}(\mathbf{P}_t; \mathcal{B}_t) = \frac{1}{2\epsilon_t}\left(\mathcal{L}\left(\mathbf{P}_t + \epsilon_t\mathbf{Z}_t; \mathcal{B}_t\right) - \mathcal{L}\left(\mathbf{P}_t - \epsilon_t\mathbf{Z}_t; \mathcal{B}_t\right)\right)\mathbf{Z}_t \tag{B.4}$$

A third-order Taylor expansion of $\mathcal{L}(\mathbf{P}_t \pm \epsilon_t\mathbf{Z}_t; \mathcal{B}_t)$:

$$\mathcal{L}(\mathbf{P}_t + \epsilon_t\mathbf{Z}_t; \mathcal{B}_t) = \mathcal{L}(\mathbf{P}_t; \mathcal{B}_t) + \epsilon_t\mathbf{Z}_t^\top\nabla\mathcal{L}(\mathbf{P}_t; \mathcal{B}_t) + \frac{\epsilon_t^2}{2}\mathbf{Z}_t^\top\nabla^2\mathcal{L}(\mathbf{P}_t; \mathcal{B}_t)\mathbf{Z}_t + \frac{\epsilon_t^3}{6}D^3\mathcal{L}(\mathbf{P}_{\xi_1}; \mathcal{B}_t)[\mathbf{Z}_t, \mathbf{Z}_t, \mathbf{Z}_t] + \mathcal{O}(\epsilon_t^4), \tag{B.5}$$

$$\mathcal{L}(\mathbf{P}_t - \epsilon_t\mathbf{Z}_t; \mathcal{B}_t) = \mathcal{L}(\mathbf{P}_t; \mathcal{B}_t) - \epsilon_t\mathbf{Z}_t^\top\nabla\mathcal{L}(\mathbf{P}_t; \mathcal{B}_t) + \frac{\epsilon_t^2}{2}\mathbf{Z}_t^\top\nabla^2\mathcal{L}(\mathbf{P}_t; \mathcal{B}_t)\mathbf{Z}_t - \frac{\epsilon_t^3}{6}D^3\mathcal{L}(\mathbf{P}_{\xi_2}; \mathcal{B}_t)[\mathbf{Z}_t, \mathbf{Z}_t, \mathbf{Z}_t] + \mathcal{O}(\epsilon_t^4), \tag{B.6}$$

where $\mathbf{P}_{\xi_1}$ lies between $\mathbf{P}_t$ and $\mathbf{P}_t + \epsilon_t \mathbf{Z}_t$, and $\mathbf{P}_{\xi_2}$ lies between $\mathbf{P}_t$ and $\mathbf{P}_t - \epsilon_t \mathbf{Z}_t$. $D^3\mathcal{L}[\cdot,\cdot,\cdot]$ denotes the third-order derivative tensor of the loss function applied to three vectors.

Substituting Eqs. B.5 and B.6 into the SPSA formula, Eq. B.4, we obtain:

$$\begin{aligned}\widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t) = &\left(\mathbf{Z}_t^\top \nabla\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)\right)\mathbf{Z}_t \\ &+ \frac{\epsilon_t^2}{12}\left(D^3\mathcal{L}(\mathbf{P}_{\xi_1};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right. \\ &\left.+D^3\mathcal{L}(\mathbf{P}_{\xi_2};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right)\mathbf{Z}_t \\ &+ \mathcal{O}(\epsilon_t^4)\end{aligned} \tag{B.7}$$

Thus, the expectation of the SPSA gradient estimate with respect to $\mathbf{Z}_t$ is:

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}_t}[\widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)] &= \mathbb{E}_{\mathbf{Z}_t}\left[\left(\mathbf{Z}_t^\top \nabla\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)\right)\mathbf{Z}_t\right] \\ &\quad+ \frac{\epsilon_t^2}{12}\mathbb{E}_{\mathbf{Z}_t}\left[\left(D^3\mathcal{L}(\mathbf{P}_{\xi_1};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right.\right. \\ &\quad\left.\left.+D^3\mathcal{L}(\mathbf{P}_{\xi_2};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right)\mathbf{Z}_t\right] \\ &\quad+ \mathcal{O}(\epsilon_t^4) \\ &= \nabla\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t) \\ &\quad+ \frac{\epsilon_t^2}{12}\mathbb{E}_{\mathbf{Z}_t}\left[\left(D^3\mathcal{L}(\mathbf{P}_{\xi_1};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right.\right. \\ &\quad\left.\left.+D^3\mathcal{L}(\mathbf{P}_{\xi_2};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right)\mathbf{Z}_t\right] \\ &\quad+ \mathcal{O}(\epsilon_t^4).\end{aligned} \tag{B.8}$$

It is worth noting that the n-SPSA gradient estimator, $\widehat{\nabla}_{n\text{-SPSA}}$, is formed by averaging n independent SPSA gradient estimators. Due to the linearity of expectation, the expected value of $\widehat{\nabla}_{n\text{-SPSA}}$ is simply the average of the expected values of these individual estimators. Since each individual SPSA estimator has the same expectation as derived above, the expected value of the n-SPSA estimator retains the same form. Therefore, for the sake of brevity and to avoid redundant derivations, we only present the detailed decomposition and expectation analysis for the single-sample SPSA estimator here.

B.3. Bounding $-\eta\mathbb{E}_t[\nabla\mathcal{L}(\mathbf{P}_t)^\top \widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)]$

We first take the expectation of $\widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)$ with respect to $\mathbf{Z}_t$, and then with respect to $\mathcal{B}_t$:

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}_t}[\widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)] &= \mathbb{E}_{\mathcal{B}_t}\left[\mathbb{E}_{\mathbf{Z}_t}[\widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)]\right] \\ &= \mathbb{E}_{\mathcal{B}_t}\left[\nabla\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)\right. \\ &\quad+\frac{\epsilon_t^2}{12}\mathbb{E}_{\mathbf{Z}_t}\left[\left(D^3\mathcal{L}(\mathbf{P}_{\xi_1};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right.\right. \\ &\quad\left.\left.+D^3\mathcal{L}(\mathbf{P}_{\xi_2};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right)\mathbf{Z}_t\right] \\ &\quad\left.+\mathcal{O}(\epsilon_t^4)\right].\end{aligned} \tag{B.9}$$

Since $\mathbb{E}_{\mathcal{B}_t}[\nabla\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)] = \nabla\mathcal{L}(\mathbf{P}_t)$, we obtain:

$$\begin{aligned}\mathbb{E}_t[\widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)] &= \nabla\mathcal{L}(\mathbf{P}_t) \\ &\quad+ \frac{\epsilon_t^2}{12}\mathbb{E}_t\left[\left(D^3\mathcal{L}(\mathbf{P}_{\xi_1};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right.\right. \\ &\quad\left.\left.+D^3\mathcal{L}(\mathbf{P}_{\xi_2};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right)\mathbf{Z}_t\right] \\ &\quad+ \mathcal{O}(\epsilon_t^4)\end{aligned} \tag{B.10}$$

Let $\mathbf{b}(\mathbf{P}_t;\mathcal{B}_t)$ be the expected bias term introduced by non-zero ϵ_t, defined as:

$$\begin{aligned}\mathbf{b}(\mathbf{P}_t;\mathcal{B}_t) = &\frac{\epsilon_t^2}{12}\mathbb{E}_t\left[\left(D^3\mathcal{L}(\mathbf{P}_{\xi_1};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right.\right. \\ &\left.\left.+D^3\mathcal{L}(\mathbf{P}_{\xi_2};\mathcal{B}_t)[\mathbf{Z}_t,\mathbf{Z}_t,\mathbf{Z}_t]\right)\mathbf{Z}_t\right] \\ &+ \mathcal{O}(\epsilon_t^4).\end{aligned} \tag{B.11}$$

Then the first term in the descent lemma becomes:

$$\begin{aligned}&-\eta\mathbb{E}_t[\nabla\mathcal{L}(\mathbf{P}_t)^\top \widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)] \\ &\quad= -\eta\nabla\mathcal{L}(\mathbf{P}_t)^\top \mathbb{E}_t[\widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)] \\ &\quad= -\eta\nabla\mathcal{L}(\mathbf{P}_t)^\top\left(\nabla\mathcal{L}(\mathbf{P}_t) + \mathbf{b}(\mathbf{P}_t;\mathcal{B}_t)\right) \\ &\quad= -\eta\|\nabla\mathcal{L}(\mathbf{P}_t)\|^2 - \eta\nabla\mathcal{L}(\mathbf{P}_t)^\top \mathbf{b}(\mathbf{P}_t;\mathcal{B}_t)\end{aligned} \tag{B.12}$$

According to Assumption 1 (ℓ-smoothness), the third-order derivative of the loss function is bounded, typically controlled by a constant L_H. Simultaneously, Assumption 2 (Local Effective Rank) indicates that despite the high dimensionality d of the parameter space, the curvature and higher-order variations of the loss function are primarily concentrated in a subspace with an effective dimension $r \ll d$. Therefore, $\|\mathbf{b}(\mathbf{P}_t;\mathcal{B}_t)\|$ can be bounded proportionally to $\epsilon_t^2 \ell r$, where ℓ is the Lipschitz constant, and L_H is of the same order as ℓ. Specifically, there exists a constant C_1 such that $\|\mathbf{b}(\mathbf{P}_t;\mathcal{B}_t)\| \le C_1\epsilon_t^2\ell r$.

Therefore

$$\begin{aligned}&|-\eta\nabla\mathcal{L}(\mathbf{P}_t)^\top \mathbf{b}(\mathbf{P}_t;\mathcal{B}_t)| \\ &\quad\le \eta\|\nabla\mathcal{L}(\mathbf{P}_t)^\top\|\|\mathbf{b}(\mathbf{P}_t;\mathcal{B}_t)\| \\ &\quad\le GC_1\epsilon_t^2\ell r = C\epsilon_t^2\ell r,\end{aligned} \tag{B.13}$$

where G is an upper bound for $\|\nabla\mathcal{L}(\mathbf{P}_t)^\top\|$, and $C = GC_1$.

Thus, the first term in the descent lemma can be bounded as

$$\begin{aligned}&-\eta\mathbb{E}_t[\nabla\mathcal{L}(\mathbf{P}_t)^\top \widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)] \\ &\quad\le -\eta\|\nabla\mathcal{L}(\mathbf{P}_t)\|^2 + C\epsilon_t^2\ell r\end{aligned} \tag{B.14}$$

B.4. Bounding $\frac{\eta^2\ell}{2}\mathbb{E}_t[\|\widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)\|^2]$

For the n-SPSA estimator, the expectation of its squared norm can be bounded as [9]:

Time budget	0	100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
FOZO (base)	59.5	59.6	60.0	60.6	60.9	61.3	61.7	62.0	62.3	62.5	62.8	63.0	63.2	63.3	63.4
FOA	58.0	59.0	60.5	61.6	62.4	63.1	63.6	64.1	64.4	64.7	64.9	65.1	65.4	65.5	65.7
ZOA	58.7	59.7	61.2	62.5	63.4	63.8	64.1	64.3	64.6	64.8	65.0	65.1	65.2	65.3	65.4
FOZO	**59.5**	**59.8**	**61.7**	**63.0**	**63.7**	**64.3**	**64.8**	**65.2**	**65.5**	**65.7**	**66.0**	**66.2**	**66.3**	**66.4**	**66.6**

Table C.1: Detailed data on the time budget and accuracy comparison of Forward-Only Test-Time Adaptation Algorithms.

$$\mathbb{E}_t[\|\widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)\|^2] \leq \gamma\mathbb{E}[\|\nabla\mathcal{L}(\mathbf{P}_t;\mathcal{B})\|^2] \tag{B.15}$$

where $\gamma = \Theta(r/n)$ is a factor related to the effective rank. Despite the potentially large parameter dimension d, due to the low effective rank property of the loss function landscape, the convergence rate of zeroth-order methods does not slow down proportionally to d as in classical analysis, but rather proportionally to r.

Therefore, the second term in the descent lemma is bounded as:

$$\begin{aligned}\frac{\eta^2\ell}{2}&\mathbb{E}_t[\|\widehat{\nabla}\mathcal{L}(\mathbf{P}_t;\mathcal{B}_t)\|^2] \\ &\leq \frac{\eta^2\ell}{2}\gamma\mathbb{E}[\|\nabla\mathcal{L}(\mathbf{P}_t;\mathcal{B})\|^2]\end{aligned} \tag{B.16}$$

B.5. Integrating into the Descent Lemma

Substituting the bounds for the bias term and variance term back into the initial expression of the descent lemma Eqn.B.3:

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\mathbf{P}_{t+1})] - \mathcal{L}(\mathbf{P}_t) \leq &-\eta\|\nabla\mathcal{L}(\mathbf{P}_t)\|^2 \\ &+ C\eta\ell\epsilon_t^2 r \\ &+ \frac{\eta^2\ell}{2}\gamma\mathbb{E}[\|\nabla\mathcal{L}(\mathbf{P}_t;\mathcal{B})\|^2].\end{aligned} \tag{B.17}$$

This completes the proof of Theorem 1.

B.6. Convergence

The variance of the gradient noise introduced by randomly arriving mini-batches is bounded, i.e., $\mathbb{E}[\|\nabla\mathcal{L}(\mathbf{P}_t;\mathcal{B})\|^2] \leq \sigma^2$. Therefore, we obtain:

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\mathbf{P}_{t+1})] - \mathcal{L}(\mathbf{P}_t) \leq &-\eta\|\nabla\mathcal{L}(\mathbf{P}_t)\|^2 \\ &+ C\eta\ell\epsilon_t^2 r + \frac{\eta^2\ell\gamma\sigma^2}{2}.\end{aligned} \tag{B.18}$$

Summing the above inequality from $t = 0$ to $T-1$, where T is the number of iterations over a period of time:

$$\begin{aligned}\sum_{t=0}^{T-1}\eta\mathbb{E}[\|\nabla\mathcal{L}(\mathbf{P}_t)\|^2] \leq &\sum_{t=0}^{T-1}\mathbb{E}[\mathcal{L}(\mathbf{P}_t)] \\ &- \sum_{t=0}^{T-1}\mathbb{E}[\mathcal{L}(\mathbf{P}_{t+1})] \\ &+ \sum_{t=0}^{T-1}\left(C\eta\ell\epsilon_t^2 r + \frac{\eta^2\ell\gamma\sigma^2}{2}\right)\end{aligned} \tag{B.19}$$

$$\begin{aligned}\eta\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\mathbf{P}_t)\|^2] \leq &\mathcal{L}(\mathbf{P}_0) - \mathbb{E}[\mathcal{L}(\mathbf{P}_T)] \\ &+ T\left(C\eta\ell\epsilon_t^2 r + \frac{\eta^2\ell\gamma\sigma^2}{2}\right)\end{aligned} \tag{B.20}$$

The loss cannot decrease indefinitely, meaning $\mathcal{L}(\mathbf{P}_T) \geq \mathcal{L}^*$. Thus, we have $\mathbb{E}[\mathcal{L}(\mathbf{P}_T)] \geq \mathcal{L}^*$:

$$\begin{aligned}\eta\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\mathbf{P}_t)\|^2] \leq &\mathcal{L}(\mathbf{P}_0) - \mathcal{L}^* \\ &+ T\left(C\eta\ell\epsilon_t^2 r + \frac{\eta^2\ell\gamma\sigma^2}{2}\right)\end{aligned} \tag{B.21}$$

Within T iterations, the upper bound for the expected average squared gradient norm is:

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[|\nabla\mathcal{L}(\mathbf{P}_t)|^2] \leq &\frac{\mathcal{L}(\mathbf{P}_0) - \mathcal{L}^*}{T\eta} \\ &+ \underbrace{C\ell\epsilon_t^2 r + \frac{\eta\ell\gamma\sigma^2}{2}}_{\text{Error Floor}}\end{aligned} \tag{B.22}$$

As the number of iterations $T \to \infty$, the first term on the right-hand side $\frac{\mathcal{L}(\mathbf{P}_0)-\mathcal{L}^*}{T\eta} \to 0$. This implies that the expected average squared gradient norm of the algorithm does not diverge, but is instead bounded by an 'Error Floor':

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}(\mathbf{P}_t)\|^2] \leq C\ell\epsilon_t^2 r + \frac{\eta\ell\gamma\sigma^2}{2} \tag{B.23}$$

The algorithm will eventually enter and remain within the neighborhood of a stationary point (a point where the gradient is zero). The size of this neighborhood is controlled by η and ϵ_t.

C. More Details on Figure 1

In this section, we present the detailed experimental results for each method shown in Figure 1 of the main paper. Table C.1 present the detailed experimental results for all methods shown in Figure 1. All results were tested on ImageNet-C (level 5). To evaluate the convergence rate of different methods across all corruptions, we reset the model parameters at each domain switch. We recorded the ACC@1 for

	α=0.1	α=0.2	α=0.3	α=0.4	α=0.5	α=0.6	α=0.7	α=0.8	α=**0.9**
ACC@1	59.13	59.15	59.04	59.34	59.33	59.45	59.47	59.50	**59.52**

Table D.1: Effects of the decay factor α in Eqn. 12. We report the continual adaptation results on ImageNet-C (5k,level 5) with 2 forward passes.

	β =0.1	β =0.2	β =0.3	β =0.4	β =0.5	β =0.6	β =0.7	β =0.8	β =**0.9**
ACC@1	59.02	59.05	59.14	59.23	59.27	59.26	59.38	59.50	**59.52**

Table D.2: Effects of the moving average factor β in Eqn. 12. We report the continual adaptation results on ImageNet-C (5k,level 5) with 2 forward passes.

	λ =0.1	λ =0.2	λ =0.3	λ =**0.4**	λ =0.5	λ =0.6	λ =0.7	λ =0.8	λ =0.9
ACC@1	59.31	59.30	59.35	**59.52**	59.50	59.34	59.30	59.27	59.16

Table D.3: Effects of trade-off parameter λ in Eqn. 15. We report the continual adaptation results on ImageNet-C (5k,level 5) with 2 forward passes.

each method adapting to each corruption under the same time budget. The accuracy corresponding to the time points listed in the table is the average value across 15 corruptions under the current time budget. As evident from Table C.1 and Figure 1, FOZO achieves higher accuracy, consistently outperforming the state-of-the-art forward-only propagation methods ZOA and FOA across all different time budgets. This indicates that our dynamic perturbation method can accelerate the convergence speed, achieving higher accuracy in a shorter amount of time.

D. More Ablation Studies

To evaluate FOZO's sensitivity to key hyperparameters, we conducted further ablation studies. We first investigated the effect of the decay factor α in Eqn. 12 (see Table 1). The results indicate that as α decreases, the perturbation scale decays faster, which affects FOZO's exploration capability during the early stages of adaptation. Next, we analyzed the impact of the moving average factor β in Eqn. 12 (see Table 2), finding that when β is significantly small, the performance degrades, attributed to the generation of biased objectives. Finally, we examined the influence of the trade-off parameter λ in Eqn. 15 (see Table 3), and ultimately selected λ=0.4 to achieve optimal performance.

E. More Experiment: Mixed Shifts

To further evaluate the robustness of FOZO under more diverse experimental settings, we introduce the mixed shifts scenario. This setup simulates a highly dynamic and unpredictable real-world scenario where the test data stream consists of samples from multiple, randomly interleaved domain shifts. Similar to the 'Continual Adaptation' setup, we utilize the ImageNet-C (5k) dataset for this scenario.

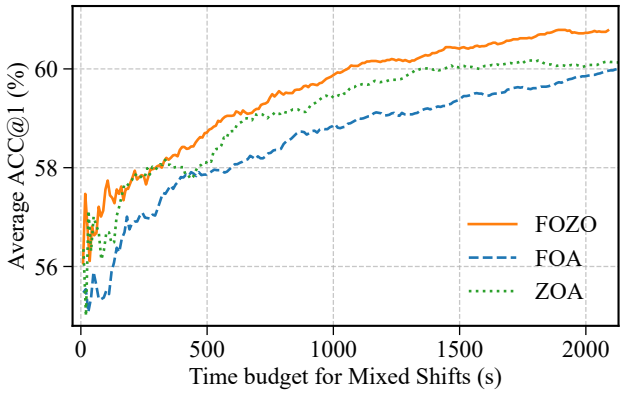

Figure E.1. Mixed shift: Performance comparison on ImageNet-C (5K, level 5).

This means that the data stream is composed of 5,000 randomly sampled images from various corruption types within ImageNet-C, presented in an unsorted, mixed fashion. This evaluates the method's robustness and agility in handling simultaneous and unannounced distribution changes with limited data per domain, reflecting a more realistic and challenging online adaptation environment.

Figure E.1 compares the performance of FOZO, ZOA, and FOA on ImageNet-C (5K, level 5) under the challenging mixed shift scenario. All three methods show increasing accuracy with a larger time budget, indicating successful adaptation. Crucially, FOZO consistently outperforms the state-of-the-art forward-only prompt tuning method FOA throughout the process and surpasses ZOA's final accuracy. These results demonstrate FOZO's enhanced robustness and superior adaptation capability in highly dynamic mixed shift environments.

References

[1] Malik Boudiaf, Romain Müller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *CVPR*, pages 8334–8343. IEEE, 2022. 1

[2] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 2

[3] Zeshuai Deng, Guohao Chen, Shuaicheng Niu, Hui Luo, Shuhai Zhang, Yifan Yang, Renjie Chen, Wei Luo, and Mingkui Tan. Test-time model adaptation for quantized neural networks. In *Proceedings of the 33rd ACM International Conference on Multimedia*, page 7258–7267, New York, NY, USA, 2025. Association for Computing Machinery. 1

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021. 1

[5] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR (Poster)*. OpenReview.net, 2019. 2

[6] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pages 8320–8329. IEEE, 2021. 2

[7] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *NeurIPS*, pages 2427–2440, 2021. 1

[8] 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. 1

[9] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *NeurIPS*, 2023. 3

[10] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, pages 16888–16905. PMLR, 2022. 1

[11] 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 *ICLR*. OpenReview.net, 2023. 1

[12] Shuaicheng Niu, Chunyan Miao, Guohao Chen, Pengcheng Wu, and Peilin Zhao. Test-time model adaptation with only forward passes. In *ICML*. OpenReview.net, 2024. 1

[13] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*. OpenReview.net, 2021. 1

[14] Haohan Wang, Songwei Ge, Zachary C. Lipton, and Eric P. Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, pages 10506–10518, 2019. 2

[15] Ross Wightman. Pytorch image models, 2019. 1

[16] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *ECCV (12)*, pages 191–207. Springer, 2022. 1