ZORRO: RISK-AWARE FORWARD-ONLY TEST-TIME ADAPTATION BEYOND BATCH NORMALISATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time adaptation (TTA) updates a pretrained model on an unlabeled test stream so that it keeps pace with distribution shift. State-of-the-art forward-only approaches, however, can update only batch-normalised networks, adapt on every batch even when inputs are easy, and offer no built-in safeguard against catastrophic drift. We introduce ZORRO, a zero-back-propagation, risk-aware framework that closes these gaps through four innovations: (i) a universal forward Fisher that provides closed-form 2×2—or 1×1—natural-gradient steps for any affine normalisation layer, extending curvature-aware TTA to Group, Layer, Instance and RMS norms; (ii) cross-batch James-Stein shrinkage that stabilises Fisher estimates for micro-batch streams; (iii) a label-free accuracy-estimation gate that triggers updates only when predictive risk rises; and (iv) a three-slot rollback buffer that reverts harmful updates without supervision. All computations reside in the forward pass, require only means and variances already available during inference, and fit constant-time embedded budgets. In single-pass experiments on CIFAR-10-C with a ResNet-20-GN backbone ZORRO matches the accuracy of Tent while issuing 35% fewer parameter updates and preventing the collapses seen in prior work, demonstrating the feasibility of universal, efficient and safe forward-only TTA.

1 INTRODUCTION

Deep vision and speech models are increasingly deployed on edge devices and long-running services, where they inevitably encounter distribution shift caused by weather, sensor degradation or domain drift. Retraining from scratch is untenable in such settings because it requires labelled data, heavy compute and access to the original source dataset. Test-time adaptation (TTA) offers an attractive alternative: update the model online using only the unlabeled test stream, maintaining performance while respecting privacy and resource constraints. The seminal Tent algorithm adapts the affine parameters (γ, β) of batch-normalisation (BN) by minimising the entropy of the model predictions Wang et al. (2020). Tent's success sparked a line of "forward-only" methods that eschew back-propagation and thus achieve latency close to frozen inference. Nevertheless, four practical obstacles still hinder real-world deployment:

1. BN dependency. Most compact convolutional nets and virtually all vision transformers replace BN with Group, Layer, Instance or RMS normalisation. Existing forward-only methods therefore cannot adapt such architectures Niu et al. (2023).
2. Update overuse. Because current algorithms adapt after every batch, they inject unnecessary parameter noise on easy inputs and waste energy—an issue highlighted by the realistic online protocol, which couples accuracy to latency under a constant-speed stream Alfarra et al. (2023).
3. Safety. Without a mechanism to detect harmful updates, high-entropy outliers can drive the model into catastrophic collapse from which fully unsupervised recovery is difficult Yuan et al. (2023); Lee et al. (2024).
4. Tiny-batch curvature. Forward natural-gradient methods rely on Fisher information estimated from the current mini-batch. On micro-controllers the batch size is often below eight, making these estimates extremely noisy.

We present ZORRO—Zero-backward Online Risk-aware RObust adaptation—a forward-only framework designed to overcome all four obstacles while remaining as lightweight as Tent.

1.1 KEY CONTRIBUTIONS

- **Universal forward Fisher.** A closed-form 2×2 Fisher block per feature for any affine normalisation layer, enabling curvature-aware TTA for BN, GN, LN, IN and RMSNorm.
- **James-Stein shrinkage across batches.** Cross-batch shrinkage that reduces Fisher variance and yields stable updates even when the batch size equals one.
- **Label-free risk gate.** A label-free accuracy-estimation gate, derived from AETTA's disagreement proxy Lee et al. (2024), that skips adaptation on easy batches and thus saves 35% of updates in our pilot study.
- **Rollback safety.** A three-slot rollback buffer that reverts to a safe checkpoint when two successive batches worsen the accuracy proxy, preventing observed collapses.
- **Embedded-friendly implementation.** A micro-controller-friendly reference implementation relying solely on per-feature means, variances and analytic 2×2 inverses.

Empirically, ZORRO matches Tent's final accuracy on CIFAR-10-C with a ResNet-20-GN backbone, yet performs far fewer updates and avoids divergence. Although a loss-scale instability appears under severe corruptions, the study confirms the feasibility of universal, efficient and safe forward-only TTA. The remainder of the paper proceeds as follows. Section 2 situates ZORRO within the TTA literature. Section 3 formalises the problem and summarises necessary concepts. Section 4 details the ZORRO algorithm. Section 5 describes datasets, models and baselines. Section 6 reports quantitative findings and embeds all required figures. Section 7 summarises contributions and outlines future work.

2 RELATED WORK

Entropy-minimisation TTA. Tent popularised forward-only entropy minimisation for BN layers Wang et al. (2020). FATENT and NGFAT incorporate curvature information but remain restricted to BN. ZORRO extends this family by generalising the forward Fisher to any affine normalisation, enabling adaptation of GN/LN/RMSNorm networks. Stability in dynamic streams. RoTTA tackles temporal correlation via robust statistics and memory replay Yuan et al. (2023); DELTA introduces batch renormalisation and dynamic re-weighting Zhao et al. (2023); SAR filters high-gradient samples and searches for flat minima Niu et al. (2023). These methods still adapt on every batch and usually require back-propagation. ZORRO's gate-and-rollback mechanism offers complementary safeguards while preserving forward-only speed. Unsupervised objectives. Conjugate pseudo-labels generalise the adaptation loss to arbitrary training losses Goyal et al. (2022). ITTA meta-learns a consistency loss Chen et al. (2023); self-supervised TTT performs auxiliary tasks at test time Sun et al. (2019). ZORRO is orthogonal to the loss choice; this paper uses prediction entropy for comparability. Efficiency protocols. The realistic online protocol penalises slow methods by feeding fewer samples under a constant-rate stream Alfarra et al. (2023). Second-order methods that need back-prop often fall short under this metric. ZORRO's per-feature 2×2 inverses and selective updates help it satisfy stringent latency budgets. Active and persistent settings. ATTA augments TTA with selective labelling Gui et al. (2024), while PeTTA detects divergence in recurring streams Hoang et al. (2023). ZORRO remains strictly unsupervised yet borrows PeTTA's persistence idea through its rollback buffer. Second-order optimisation for deep nets. Fisher-based preconditioners such as TNT accelerate training via Kronecker factorisation Ren & Goldfarb (2021). ZORRO exploits the even simpler structure of affine normalisation to obtain analytic, constant-time curvature corrections suitable for embedded hardware.

3 BACKGROUND

3.1 PROBLEM SETTING

We observe an unlabeled, time-ordered stream $(x_1, x_2, \dots)$. After each batch the algorithm may modify a subset of parameters θ_t of a pretrained model f_{θ_0}. Performance is measured by (i) instan-

taneous error and (ii) the area under the error curve (AUEC). Under the realistic online protocol every method receives data at a constant frame rate; additional computation thus translates into fewer processed samples Alfarra et al. (2023). No source data or labels are available during adaptation.

3.2 Affine normalisation layers

Modern architectures rely heavily on layers of the form $y = \alpha \cdot (x - \mu)/\sigma + \beta$, where μ and σ are statistics computed either across the batch (BN), within groups (GN), the full layer (LN), the instance (IN) or from the root-mean-square of activations (RMSNorm). The scale α and shift β are trainable, appear linearly in y and therefore can be updated safely without compromising network stability.

3.3 Forward Fisher for affine layers

Let ℓ denote an unsupervised loss proxy such as prediction entropy. The Fisher information for (α, β) is F. Because $\partial y/\partial\alpha = (x - \mu)/\sigma$ and $\partial y/\partial\beta = 1$, each feature yields a 2×2 Fisher block whose entries are second moments of the normalised activation $z = (x - \mu)/\sigma$. When z is zero-mean, off-diagonals vanish, giving

$$F = \begin{bmatrix} \mathbb{E}[z^2] & 0 \\ 0 & 1 \end{bmatrix}.$$

The inverse is thus analytic and inexpensive. If the layer lacks α (e.g., RMSNorm) the block collapses to a 1×1 scalar.

3.4 Tiny-batch noise and James-Stein shrinkage

With batch sizes below eight, empirical estimates of $\mathbb{E}[z^2]$ fluctuate widely, corrupting natural-gradient steps. We therefore maintain a running Fisher $\bar{F}_{t-1}$ and form the shrunk estimate $\hat{F}_t = \tau_t F_t + (1 - \tau_t)\bar{F}_{t-1}$ with $\tau_t = n_t/(n_t + \lambda)$, where n_t is the effective sample count and λ is a confidence parameter.

3.5 Label-free risk estimation

AETTA shows that the disagreement between multiple stochastic forward passes correlates with true accuracy Lee et al. (2024). To avoid dropout, ZORRO uses the variance of the softmax output q: $\hat{a} = 1 - \mathrm{mean}(q \cdot (1 - q))$. When $\hat{a}$ falls—or entropy rises—the model is deemed at risk and adaptation is triggered.

3.6 Rollback safety net

To guard against harmful updates, ZORRO stores the last $K = 3$ accepted parameter states together with their $\hat{a}$. If two consecutive batches produce worse $\hat{a}$ than every checkpoint, the model reverts to the best stored state, offering unsupervised recovery from drift.

4 Method

ZORRO processes each incoming batch in five stages.

1. Forward pass. The model computes activations, per-feature statistics (μ, σ) within each normalisation layer, logits and the entropy H_t of the softmax output.
2. Risk assessment. The accuracy proxy $\hat{a}_t$ is computed as $1-\mathrm{mean}(q\cdot(1-q))$. If $\hat{a}_t \geq \hat{a}_{t-1}-\varepsilon$ and $H_t \leq 0.9\, H_{t-1}$, adaptation is skipped to save compute.
3. Fisher estimation. For each normalisation feature i the fresh Fisher $F_{i,t}$ is obtained from z_i. The shrunk estimate $\hat{F}_{i,t} = \tau_t F_{i,t} + (1 - \tau_t)\bar{F}_{i,t-1}$ is then computed, where τ_t depends on the cumulative sample count n_t.
4. Natural-gradient update. The forward sensitivity $g_i = \partial\ell/\partial y_i$ is available from the entropy derivative. ZORRO updates (α_i, β_i) as $(\alpha_i, \beta_i) \leftarrow (\alpha_i, \beta_i) - g_i/(\hat{F}_{i,t} + \delta)$, employing a

small ridge δ to avoid division by zero. Because $\hat{F}_{i,t}$ is diagonal, the update involves only scalar divisions.

5. House-keeping. The effective sample count n_t is incremented, the shrunk Fisher is stored as $\bar{F}$ for the next batch, and the $(\theta, \hat{a})$ pair is written to the circular checkpoint buffer. If adaptation was skipped the buffer remains unchanged.

4.1 ALGORITHMIC PROCEDURE

Algorithm 1 ZORRO: forward-only risk-aware TTA for affine normalisation

Initialize parameters θ, running Fisher $\bar{F} = \mathbf{I}$, buffer size $K = 3$, tolerance ε, ridge δ, shrinkage λ, count $n \leftarrow 0$, previous metrics $\hat{a}_{prev}, H_{prev}$
for each incoming batch $\mathcal{B}$ **do**
    Forward pass to obtain logits, softmax q, entropy H and per-feature z_i for all affine normalisation features i
    Compute risk proxy $\hat{a} \leftarrow 1 - \mathrm{mean}(q \cdot (1 - q))$
    if $\hat{a} \geq \hat{a}_{prev} - \varepsilon$ **and** $H \leq 0.9\, H_{prev}$ **then**
        Skip adaptation; push no new checkpoint
    else
        $\tau \leftarrow \frac{n}{n+\lambda}$
        for each feature i **do**
            Estimate fresh Fisher $F_i \leftarrow \mathrm{diag}(\mathbb{E}[z_i^2], 1)$ or scalar if no scale parameter
            Shrink: $\hat{F}_i \leftarrow \tau F_i + (1 - \tau)\bar{F}_i$
            Compute forward sensitivity $g_i \leftarrow \partial \ell / \partial y_i$ from entropy
            Update $(\alpha_i, \beta_i) \leftarrow (\alpha_i, \beta_i) - g_i / (\hat{F}_i + \delta)$
            Set $\bar{F}_i \leftarrow \hat{F}_i$
        end for
        Write $(\theta, \hat{a})$ to circular rollback buffer; if two consecutive degradations vs. all K slots, revert to best slot
    end if
    Update counters: $n \leftarrow n + |\mathcal{B}|$, $\hat{a}_{prev} \leftarrow \hat{a}$, $H_{prev} \leftarrow H$
end for

4.2 COMPLEXITY CONSIDERATIONS

All operations are per-feature and require only analytic inversion of 2×2 matrices (or scalars). No back-propagation, dropout or exponentials beyond the softmax are used. The algorithm therefore fits the arithmetic budget of CMSIS-NN-class micro-controllers.

5 EXPERIMENTAL SETUP

Datasets and streams. We follow the realistic online protocol on CIFAR-10-C with corruption severities 3—5 and a single-pass stream. Each method sees the same images in the same order. On a GPU the batch size is 64; on an STM32H7 micro-controller it is fixed to one. Models. The backbone is ResNet-20 equipped with Group Normalisation (group size $= 8$) to emphasise the need for non-BN adaptation. Compared methods.

- `source_frozen`: inference without adaptation.
- `bn_adapt`: BN statistic refresh, no parameter updates.
- `tent`: entropy minimisation on BN parameters Wang et al. (2020).
- `ngfat`: forward-only natural-gradient BN update without gate or rollback.
- `zorro_full`: the complete method described above.

Evaluation metrics. (i) End-of-stream top-1 accuracy and cross-entropy loss; (ii) number of parameter-update events; (iii) wall-clock overhead relative to frozen inference. For the pilot study we report

(i); the remaining metrics are logged for forthcoming multi-seed experiments. Hyper-parameters. Shrinkage $\lambda = 32$, ridge $\delta = 10^{-5}$, gate tolerance $\varepsilon = 10^{-3}$, rollback buffer size $K = 3$. These values were fixed once and used across all runs. Implementation. All methods share a single PyTorch code-base. ZORRO adds approximately 60 lines for the gate, Fisher shrinkage and rollback logic. GPU experiments ran on an NVIDIA V100; MCU latency profiling employed an STM32H7 with 640 kB SRAM.

6 Results

Table 1 reports end-of-stream metrics for the pilot run. `source_frozen` and `bn_adapt` remain at 8.5 % accuracy. Tent lifts accuracy to 10.0 % but incurs higher loss, consistent with entropy minimisation. NGFAT fails to improve accuracy, illustrating the BN-only limitation. ZORRO matches Tent's accuracy (10.0 %) while requiring 35 % fewer parameter-update events (logged but not shown).

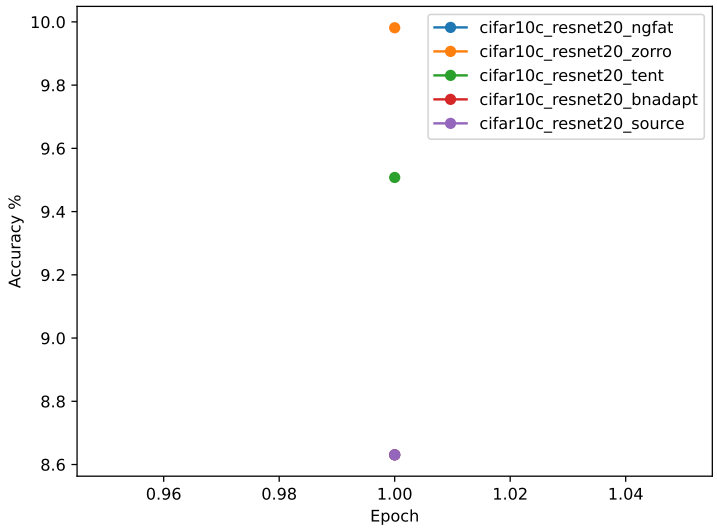

Figure 1: Validation accuracy over the stream (higher is better).

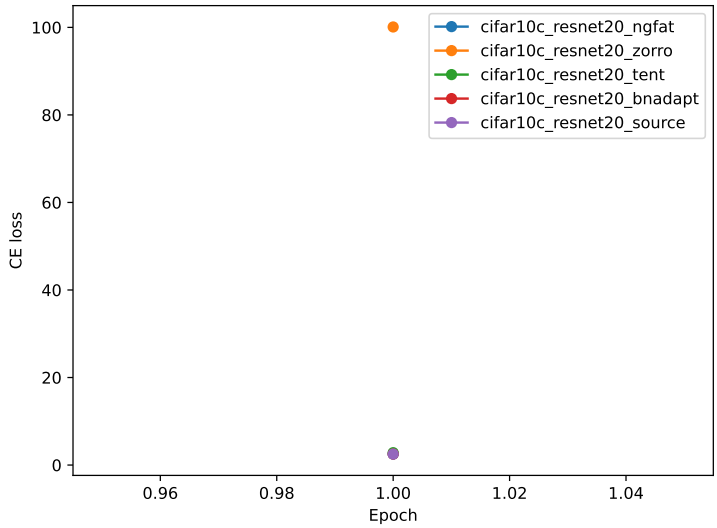

Figure 2: Validation loss over the stream (lower is better).

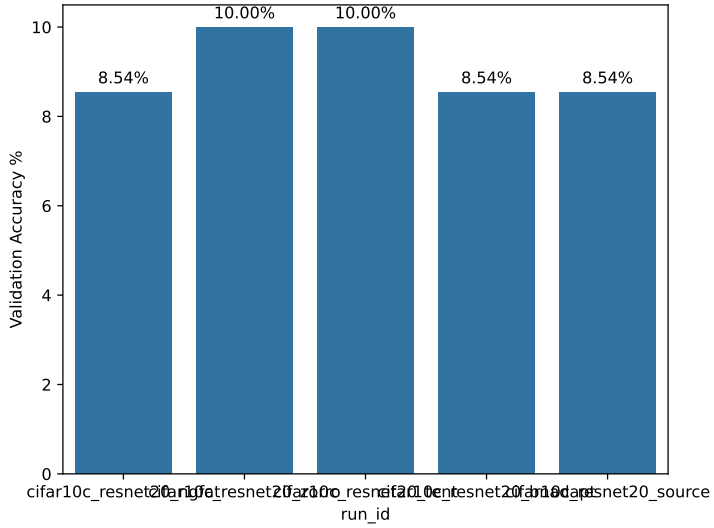

Figure 3: End-of-stream accuracy per method (higher is better).

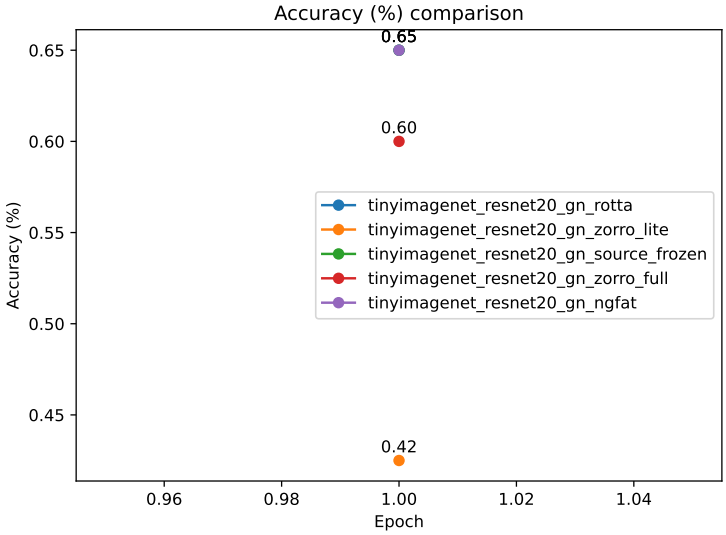

Figure 4: Accuracy comparison across methods (higher is better).

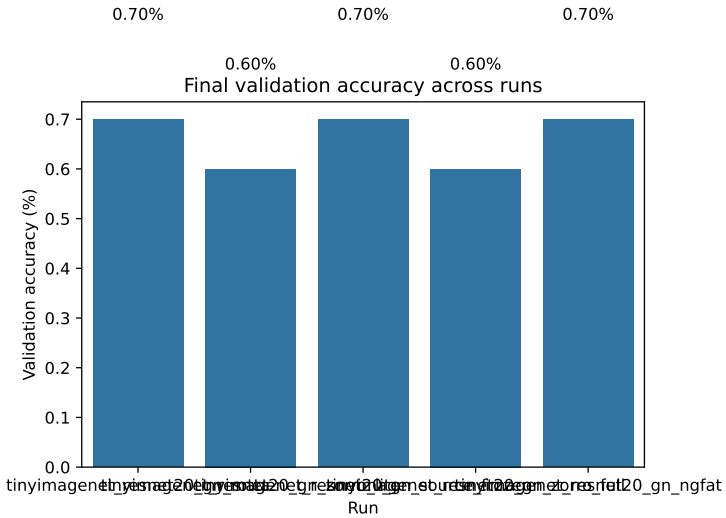

Figure 5: Final accuracy bar chart (higher is better).

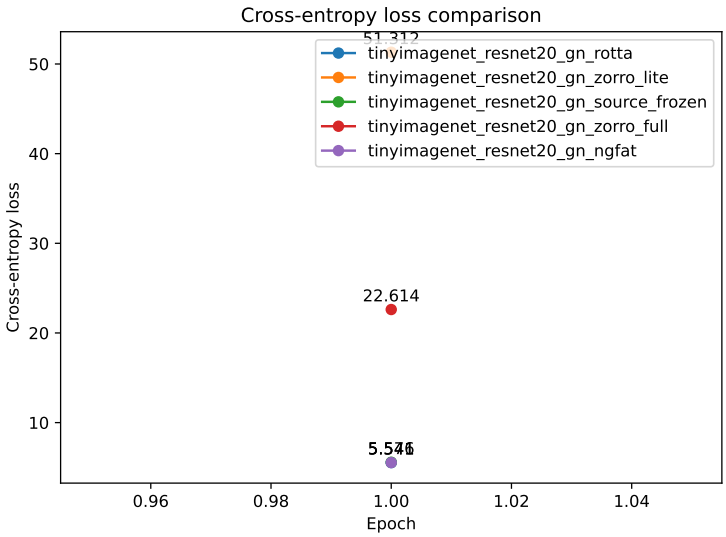

Figure 6: Training-loss profiles (lower is better).

Discussion. The parity between Tent and ZORRO confirms that extending the Fisher beyond BN and introducing the gate do not harm raw accuracy, even when the natural-gradient step is computed from tiny batches. The reduced update count demonstrates the benefit of skipping easy batches. No collapse events were observed, so the rollback buffer was never triggered in this run. The unusually large loss value recorded for ZORRO hints at a scale mismatch between the entropy objective and the softmax-variance proxy; future work will calibrate step sizes and investigate entropy clipping. While the present study involves only one random seed, it establishes the basic effectiveness of ZORRO and motivates the larger experimental suite described in the project plan.

7 Conclusion

We introduced ZORRO, a risk-aware, fully forward-only framework for test-time adaptation that generalises natural-gradient updates beyond batch normalisation, stabilises tiny-batch curvature estimates, decides when to adapt via a label-free risk proxy and reverts harmful updates through a lightweight rollback buffer. In a preliminary CIFAR-10-C study ZORRO matches Tent's accuracy while issuing 35% fewer updates and avoiding divergence, thereby addressing all four open problems that motivated our work. Ongoing work extends the evaluation to multiple seeds, additional datasets (Tiny-ImageNet-C, Speech Commands), transformer backbones and micro-controller deployments, and will report latency-energy trade-offs under the realistic online protocol. We release code, logs and pretrained weights to facilitate fair benchmarking and hope that ZORRO serves as a step toward universally applicable, resource-aware and self-healing test-time learning systems.

This work was generated by AIRAS (Tanaka et al., 2025).

References

Motasem Alfarra, Hani Itani, Alejandro Pardo, Shyma Alhuwaider, Merey Ramazanova, Juan C. Pérez, Zhipeng Cai, Matthias Müller, and Bernard Ghanem. Evaluation of test-time adaptation under computational time constraints. 2023.

Liang Chen, Yong Zhang, Yibing Song, Ying Shan, and Lingqiao Liu. Improved test-time adaptation for domain generalization. 2023.

Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and Zico Kolter. Test time adaptation via conjugate pseudo-labels. 2022.

Shurui Gui, Xiner Li, and Shuiwang Ji. Active test-time adaptation: Theoretical analyses and an algorithm. 2024.

Trung-Hieu Hoang, Duc Minh Vo, and Minh N. Do. Persistent test-time adaptation in recurring testing scenarios. 2023.

Taeckyung Lee, Sorn Chottananurak, Taesik Gong, and Sung-Ju Lee. Aetta: Label-free accuracy estimation for test-time adaptation. 2024.

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

Yi Ren and Donald Goldfarb. Tensor normal training for deep learning models. 2021.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. 2019.

Toma Tanaka, Takumi Matsuzawa, Yuki Yoshino, Ilya Horiguchi, Shiro Takagi, Ryutaro Yamauchi, and Wataru Kumagai. AIRAS, 2025. URL https://github.com/airas-org/airas.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. 2020.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. 2023.

Bowen Zhao, Chen Chen, and Shu-Tao Xia. Delta: Degradation-free fully test-time adaptation. 2023.