ZEROSIAM: AN EFFICIENT ASYMMETRY FOR TEST-TIME ENTROPY OPTIMIZATION WITHOUT COLLAPSE

Guohao Chen[1*], **Shuaicheng Niu**[1*], **Deyu Chen**[2], **Jiahao Yang**[2], **Zitian Zhang**[2]
Mingkui Tan[2*], **Pengcheng Wu**[1], **Zhiqi Shen**[1†]
Nanyang Technological University[1] South China University of Technology[2]

ABSTRACT

Test-time entropy minimization can favor non-generalizable shortcuts, such as inflating the logit norm and driving all predictions to a dominant class to reduce entropy, risking collapsed solutions (*e.g.*, constant one-hot outputs) that trivially minimize the objective without meaningful learning. In this paper, we reveal asymmetry as a key mechanism for collapse prevention and introduce ZeroSiam–an efficient asymmetric Siamese architecture tailored for test-time entropy minimization. ZeroSiam efficiently achieves asymmetry with a learnable predictor and a stop-gradient operator before the classifier. We provide empirical and theoretical evidence that ZeroSiam not only prevents collapse, but also regularizes biased learning signals, enhancing performance even when no collapse occurs. Despite its simplicity, extensive results show that ZeroSiam performs more stably over prior methods using negligible overhead, demonstrating efficacy on both vision adaptation and large language model reasoning tasks across challenging test scenarios and diverse models, including the particularly collapse-prone tiny models.

1 INTRODUCTION

Entropy measures the uncertainty of a model's predictions. Since its emergence, entropy optimization has been widely adopted as an auxiliary objective alongside supervised, reinforcement signals, *etc.* Recently, **test-time** entropy minimization has attracted growing interest, as it operates purely unsupervised during inference without relying on any ground-truth supervision, enabling adaptation to novel or out-of-distribution domains (Liang et al., 2024) and moving toward greater intelligence.

However, entropy minimization naturally drives the model to increase the maximum predicted logit, regardless of whether it aligns with the true label. In this sense, minimizing entropy alone during testing can lead to degraded performance—exploiting a shortcut that produces near one-hot outputs for all inputs, which trivially minimizes entropy but fails to capture meaningful predictions.

Various TTA methods have been proposed to tackle the above issue, such as filtering unreliable gradients using thresholds (Lee et al., 2024), or mitigating disturbances through optimization of a sharpness-aware loss surface (Niu et al., 2023). However, we highlight that existing methods remain exposed to collapse risk, as the model can still minimize entropy by collapsing into constant one-hot outputs regardless of the inputs. As in Tables 2-3, it is unavoidable that these methods demonstrate high instability in practice, particularly under prolonged or more challenging test scenarios.

In this paper, we reveal asymmetry as a key mechanism for collapse prevention, which was originally developed in the negative-free self-supervised learning (Chen & He, 2021; Zhang et al., 2022) to prevent the *two branches* from collapsing toward the same constant in consistency learning. However, naively applying asymmetry to our context is infeasible or inferior, as: 1) entropy minimization typically has only one prediction branch and optimizes entropy instead of similarity; 2) traditional Siamese design requires extra backbone passes, which impairs efficiency for our test-time learning.

Therefore, we design ZeroSiam, demonstrating how asymmetry can be efficiently implemented in test-time entropy minimization, without using augmentations, extra backbone passes, and teacher models. The core idea in ZeroSiam is to decouple a prediction into two asymmetric outputs based on

[*]Equal contribution. [†]Corresponding author.

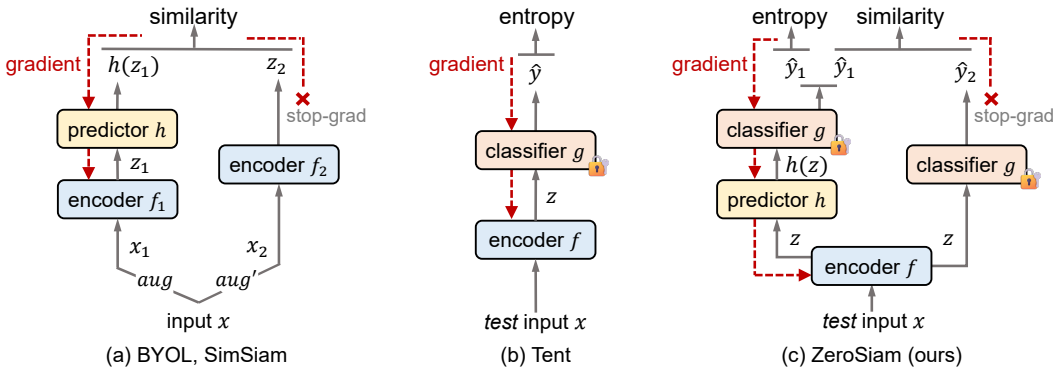

Figure 1: Comparisons on architectures. (a) Alignment-oriented SSL methods (BYOL (Grill et al., 2020), SimSiam (Chen & He, 2021)). (b) Test-time entropy minimization (Tent) (Wang et al., 2021). (c) Our ZeroSiam, which designs a minimal asymmetry for entropy minimization with a lightweight predictor and a stop-gradient branch—without augmentations, extra encoder passes, or teacher models—to substantially enhance learning stability and boost performance while retaining efficiency.

the same feature: an online branch with a learnable predictor and a target branch with stop-gradient, where we minimize entropy on the online branch to learn discrimination, while performing asymmetric predictor–target alignment to prevent collapsed constants. We investigate both the empirical and theoretical behaviors of ZeroSiam during TTA in Section 2.2, highlighting that our architecture can both avoid collapse and regularize biased learning signals at test time, which improves the performance of entropy optimization even when no collapse occurs. ZeroSiam achieves notable stability and effectiveness with a single extra predictor, requiring negligible overhead (see Table 1). Despite its simplicity, ZeroSiam achieves state-of-the-art results across diverse architectures and a wide range of test scenarios, *e.g.*, adapting reliably even with *incorrect* pseudo labels (see Table 2).

Main Contributions: 1) We are the first to study asymmetric structure in TTA and propose ZeroSiam, revealing how asymmetry can be efficiently implemented in test-time entropy minimization without requiring augmentations, extra backbone passes, and teacher models. 2) We provide theoretical and empirical insights into ZeroSiam's asymmetry for collapse prevention and bias regularization, with diverse results to demonstrate its superiority, uncovering a promising research direction.

2 ZeroSiam: Stable Adaptation via Minimal Asymmetric Network

2.1 Minimal Asymmetric Siamese Architecture for Entropy Minimization

SSL (Chen & He, 2021) prevents collapse constants in consistency learning by asymmetry: with a non-identity predictor h on one branch, and a stop-gradient operation on the other (Figure 1). However, constructing asymmetry in test-time entropy optimization remains an open question due to its single-branch nature, non-pairwise objective, and efficiency requirements for online update.

To bridge these gaps, our key idea is to embed asymmetry within a single forward pass by inserting a predictor and the stop-gradient operator before the classifier, which decouples a prediction into two *asymmetric* views from the same feature: an online branch (through the predictor) for *optimizing entropy*, and a target branch (the original logits) for *asymmetric stop-gradient alignment*. Let $f(\cdot;\theta_f)$ denote the encoder, $g(\cdot;\theta_g)$ the classifier, and $h(\cdot;\theta_h)$ a *linear* predictor. Given a test sample $\mathbf{x}$, ZeroSiam computes the encoder feature $z = f(\mathbf{x};\theta_f)$ once, then define two asymmetric branches:

$$u^r = g(z;\theta_g), \qquad \text{(target branch, stop-gradient)} \tag{1}$$

$$u^o = g(h(z;\theta_h);\theta_g), \qquad \text{(online branch)}. \tag{2}$$

Let $p^r = \mathrm{softmax}(u^r)$ and $p^o = \mathrm{softmax}(u^o)$. The ZeroSiam objective combines entropy minimization on the online branch with an alignment regularizer to the target branch:

$$\mathcal{L} = H(p^o) + \alpha\, D(p^o \,\|\, \mathrm{sg}[p^r])\,, \tag{3}$$

where $H(p) = -\sum_c p_c \log p_c$ is the prediction entropy, $D(\cdot\|\cdot)$ is a divergence (*e.g.*, ℓ_2 or KL), and $\mathrm{sg}[\cdot]$ denotes stop-gradient. Here, θ_h is initialized as an identity to ensure a warm start, which

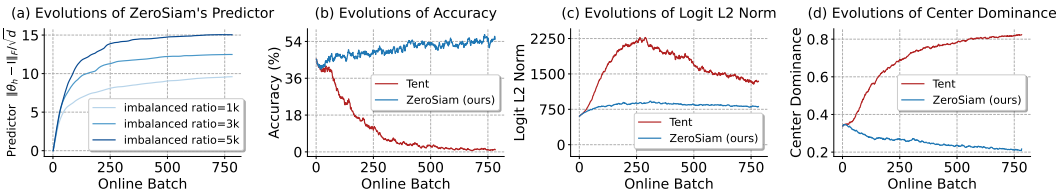

Figure 2: Empirical evidence of ZeroSiam's stabilization effects. (a) records the Frobenius distance between θ_h and the identity matrix. (b-d) record the ODD accuracy, logits L_2 norm, and center dominance (Zhang et al., 2022) during TTA. Experiments are on ImageNet-C (Snow, level 5) with ResNet50-GN. For fair comparisons, ZeroSiam and Tent use the same learning rate configuration.

quickly diverges during online learning, as shown in Figure 2 (a), creating the asymmetry necessary to prevent collapsed constant solutions. Even a *randomly initialized* predictor, Table 4 shows that ZeroSiam helps prevent collapse in Tent (Wang et al., 2021), revealing asymmetry's distinct value.

2.2 How ZeroSiam Resists Collapse: Empirical and Theoretical Insights

We empirically and theoretically find that ZeroSiam not only inherits the anti-collapse effect of asymmetry, but also absorbs bias and regularizes learning dynamics during entropy minimization.

Observation 1: Predictor as an Effective Absorber of Biased Learning Signals. Figure 2 (a) shows that θ_h deviates more rapidly, substantially from identity under an imbalanced data stream with more biased signals. This pronounced deviation implies an active absorption of biased signals in h, posing stronger alignment regularization in Eqn. (3) to stabilize TTA under more challenging scenarios.

Observation 2: Asymmetry Suppresses Learning Non-Generalizable Collapse Modes. In Figure 2 (c), the logit L_2 norm under Tent suffers from a rapid inflation, whereas ZeroSiam grows slowly and stabilizes. In Figure 2 (d), Tent increasingly aligns logits toward a dominant mode (higher center dominance), while our ZeroSiam suppresses such dominance over time. Thus, Tent collapses into non-generalizable shortcuts, whereas ZeroSiam mitigates them and achieves better generalization.

Overall, Figure 2 shows that ZeroSiam absorbs and regulates non-generalizable shortcuts during TTA beyond collapse prevention. This reveals a distinct advantage of ZeroSiam, and provides TTA-specific evidence on the role of asymmetry in generalization that extends prior findings from SSL.

Theoretical Insights of ZeroSiam's Mechanism We further provide a theoretical analysis to explain ZeroSiam's behaviors, showing (i) the predictor enhances the online branch's dynamism in exploring the parameter space, thereby facilitating more efficient entropy optimization; (ii) the alignment term explicitly regulates the biased learning signals induced by entropy minimization, preventing the model from being exposed to collapsed trivial solutions meanwhile prompting performance.

Theorem 1. ***(Optimization and Stability of ZeroSiam)*** *Consider the ZeroSiam objective* $\mathcal{L} = H(p^o) + \alpha\, D(p^o \,\|\, \mathrm{sg}[p^r])$, *where* $H(\cdot)$ *denotes the entropy loss,* $D(\cdot)$ *the alignment regularizer, and* $p^o, p^r \in \Delta^{|\mathcal{C}|-1}$ *are the probability distributions induced by the online and target branches. Given the encoder* f, *classifier* g *and predictor* h. *Under Assumptions 1 and 2, the following hold: (1) For* $\alpha = 0$, *the entropy variation satisfies* $|\Delta H(p^o)| > |\Delta H(p^r)|$, *and the Hessian of* $H(p^o)$ *attains its minimal eigenvalue along collapse directions* $\mathbf{v}$*:*

$$\lambda_{\min}\big(\nabla^2 H(p^o)\big) = \mathbf{v}^\top \nabla^2 H(p^o)\mathbf{v}.$$

(2) For $\alpha > 0$, *the predictor* h *serves as a filtering mechanism that suppresses gradient update directions corresponding to over-amplified logits, and the system converges to a stable equilibrium:*

$$\exists\, h_{\min} > 0 \quad \textit{s.t.} \quad H(p^o) > h_{\min}, \quad p^o \to p^r.$$

Remark 1. *Theorem 1 characterizes ZeroSiam's optimization dynamics and convergence behavior. Initially, the online and target branches coincide within the non-collapse region, so entropy minimization dominates. In this regime, the online branch exhibits a stronger tendency toward collapse than the target branch in both magnitude and directional dominance (**Conclusion (1)**). As optimization proceeds, the alignment term* $D(\cdot)$ *constrains the predictor* h *to filter/suppress the gradient components that drive* p^o *away from* p^r, *i.e., regulating the biased learning signals that may result in collapse, guiding the system toward a stable and non-collapsing equilibrium (**Conclusion (2)**).*

Table 2: Comparisons with SOTAs on ImageNet-C (level 5)'s **BLIND-SPOT SUBSET** (samples initially misclassified by the NoAdapt model) with **BATCH SIZE=1** regarding **Accuracy (%)**. RED marks results worse than NoAdapt. **Detailed results of each corruption are put in Appendix.**

	ResNet50-GN				ViT-Base				ViT-Small				ConvNeXt-Tiny				Swin-Tiny				
Method	$\mathcal{N}$	$\mathcal{B}$	$\mathcal{W}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{B}$	$\mathcal{W}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{B}$	$\mathcal{W}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{B}$	$\mathcal{W}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{B}$	$\mathcal{W}$	$\mathcal{D}$	Avg.
NoAdapt	18.6	19.4	47.7	33.9	8.1	28.4	36.1	41.7	1.8	23.3	27.8	33.6	23.7	24.6	52.2	35.3	25.9	21.6	50.6	25.8	29.0
Tent	0.2	5.2	16.8	28.1	0.2	42.9	20.0	52.0	0.1	40.0	36.3	42.8	17.9	19.7	36.3	31.0	0.2	8.0	22.2	13.0	21.7
SAR	17.9	18.4	35.6	31.6	31.1	55.0	54.0	56.4	2.0	30.4	42.0	43.3	26.4	24.6	50.7	35.1	27.0	22.2	49.3	23.8	33.8
EATA	19.5	19.0	44.2	40.2	26.3	43.5	53.2	57.5	2.0	26.5	32.8	41.7	25.8	24.7	51.6	35.7	27.4	22.4	49.1	26.8	33.5
COME	18.6	19.4	47.6	33.8	0.1	44.8	70.9	69.7	1.0	37.0	55.9	46.2	41.4	16.4	31.1	36.1	40.5	9.5	25.3	26.9	33.6
DeYO	0.5	9.4	21.0	40.0	0.2	47.2	72.6	71.4	0.2	38.2	60.2	61.7	35.2	12.2	35.0	33.0	0.2	8.8	25.7	23.9	29.8
ZeroSiam	39.9	36.6	53.9	52.7	53.0	60.4	71.6	70.2	26.1	49.8	61.8	61.1	43.3	40.3	61.9	55.0	46.5	42.1	58.8	54.7	**52.0**

Table 3: Comparisons with SOTAs on ImageNet-C (level 5) under **ONLINE LABEL SHIFTS** (imbalance ratio=∞) w.r.t. **Acc(%)**. "$\mathcal{N}$/$\mathcal{B}$/$\mathcal{W}$/$\mathcal{D}$" are short for *Noise/Blur/Weather/Digital* corruptions.

	ResNet50-GN				ViT-Base				ViT-Small				ConvNeXt-Tiny				Swin-Tiny				
Method	$\mathcal{N}$	$\mathcal{B}$	$\mathcal{W}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{B}$	$\mathcal{W}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{B}$	$\mathcal{W}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{B}$	$\mathcal{W}$	$\mathcal{D}$	$\mathcal{N}$	$\mathcal{B}$	$\mathcal{W}$	$\mathcal{D}$	Avg.
NoAdapt	18.6	19.4	47.7	33.9	8.1	28.4	36.1	41.6	1.8	23.4	27.7	33.5	23.7	24.8	52.1	35.4	25.8	21.7	50.5	25.9	29.0
Tent	2.9	14.6	27.7	38.0	22.9	54.3	41.3	64.7	0.3	40.5	38.5	48.2	22.8	22.9	50.2	34.4	13.6	19.4	34.9	25.3	30.9
SAR	35.0	24.9	47.8	40.6	46.2	55.8	61.4	65.8	1.6	42.2	46.7	52.8	35.5	24.2	43.3	38.0	29.2	19.0	39.4	26.7	38.8
EATA	27.8	20.4	42.7	34.6	35.7	47.4	61.6	51.6	16.3	42.2	54.8	54.2	35.4	29.0	54.9	42.2	35.6	32.9	53.4	41.1	40.7
COME	18.4	19.2	47.7	33.4	49.8	59.2	69.8	67.5	0.2	39.5	56.0	59.6	43.1	24.4	62.9	44.9	42.9	23.9	59.7	27.7	42.5
DeYO	43.7	23.2	54.2	54.5	48.0	56.6	71.2	69.9	0.1	41.4	59.4	60.7	36.0	19.0	43.8	34.8	41.9	26.8	56.6	46.8	44.4
ZeroSiam	43.7	41.2	62.1	57.3	52.8	60.1	71.2	69.6	26.5	50.4	62.0	60.8	43.0	40.0	62.4	53.8	45.3	42.2	59.3	53.6	**52.9**

Table 4: Ablation on predictor design in ZeroSiam on ImageNet-C (level 5) with ViT-Base under label shifts.

Method	Predictor	Acc.
Tent	-	47.3
ZeroSiam	FC	**64.1**
• Exp1	*fixed* random FC	60.7
• Exp2	FC+ReLU+FC	64.0

Table 5: ZeroSiam's efficacy when used with prior methods. Experiments are on ImageNet-C (level 5) under label shifts.

Method	ViT-B	R-50	Con-T	Avg.
EATA	49.4	31.6	40.7	40.6
+ *ZeroSiam*	57.1	48.3	41.7	49.0
DeYO	62.3	43.9	33.2	46.5
+ *ZeroSiam*	65.0	50.7	46.1	53.9

Table 6: Generality of ZeroSiam across objective and divergence choices on ImageNet-C (level 5) with ViT-Base under label shifts.

Loss	N/A	sKL	KL	rKL	JS	MSE
SLR	44.0	62.2	63.2	61.0	63.4	61.5
CE	43.2	61.6	61.4	61.9	61.4	62.1
$-p^2$	45.8	63.3	63.4	62.1	63.0	63.6
Tent	47.3	64.1	64.0	64.1	63.8	64.1

3 COMPARISONS WITH STATE-OF-THE-ARTS

We compare with prior SOTAs w.r.t. stability, efficiency, efficacy, and demonstrate the generality of the proposed ZeroSiam (Tables 1-6). From the results, ZeroSiam achieves a lower latency compared to prior SOTAs (Tables 1, 280s→193s compared with DeYO), meanwhile significantly improving their stability (Table 2-3 & 5). In particular, ZeroSiam delivers consistent accuracy gains *even when adapting with incorrect labels* (Table 2), demonstrating effectiveness across diverese models and improving the average accuracy of 33.8% (SAR) to 52.0% (ZeroSiam), which suggests that ZeroSiam can be safely deployed in more challenging real-world scenarios via a minimal architectural modification. We further examine other predictor designs in Table 4, showing that replacing the learnable predictor with a fixed, randomly initialized predictor already improves upon Tent, *i.e.*, 47.3%→60.7%, suggesting the value of breaking symmetry alone in stabilizing TTA. Moreover, Tables 5-6 show that ZeroSiam can generalize to different TTA methods and diverse objectives. Overall, these results highlight ZeroSiam as a promising research direction for collapse-free TTA.

Table 1: Efficiency comparison for processing 50,000 images (Gaussian noise, level 5 on ImageNet-C) via an RTX 3090 GPU on ViT-Base.

Method	GPU time
Tent (Wang et al., 2021)	193 secs
SAR (Niu et al., 2023)	382 secs
EATA (Niu et al., 2022)	197 secs
COME (Zhang et al., 2025)	300 secs
DeYO (Lee et al., 2024)	280 secs
ZeroSiam (ours)	193 secs

4 CONCLUSION

In this paper, we explore asymmetry as a principled test-time entropy minimization mechanism that improves stability by inherently avoiding collapsed trivial solutions. To this end, we devise a lightweight Siamese architecture for asymmetric entropy minimization (*termed* ZeroSiam), using an injected learnable predictor with negligible overhead. We empirically and theoretically demonstrate ZeroSiam's efficacy in preventing collapse and regularizing biased learning signals during TTA. Despite its simplicity, extensive results verify ZeroSiam's strong performance across test scenarios.

REFERENCES

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pp. 151–160. PMLR, 2019.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Alexander Bartler, Andre Bühler, Felix Wiewel, Mario Döbler, and Bin Yang. Mt3: Meta test-time training for self-supervised test-time adaption. In *International Conference on Artificial Intelligence and Statistics*, pp. 3080–3090. PMLR, 2022.

Guohao Chen, Shuaicheng Niu, Deyu Chen, Shuhai Zhang, Changsheng Li, Yuanqing Li, and Mingkui Tan. Cross-device collaborative test-time adaptation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 122917–122951, 2024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

MAA Codeforces. American invitational mathematics examination-aime, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*, volume 35, pp. 29374–29385, 2022.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *Advances in neural information processing systems*, volume 33, pp. 21271–21284, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870. Pmlr, 2018.

Jisu Han, Jaemin Na, and Wonjun Hwang. Ranked entropy minimization for continual test-time adaptation. *arXiv preprint arXiv:2505.16441*, 2025.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, pp. 1–11, 2019.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Jinwu Hu, Zitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. In *International Conference on Machine Learning*. PMLR, 2025a.

Zixuan Hu, Yichun Hu, Xiaotong Li, Shixiang Tang, and Lingyu Duan. Beyond entropy: Region confidence proxy for wild test-time adaptation. In *International Conference on Machine Learning*, 2025b.

Hyosoon Jang, Yunhui Jang, Sungjae Lee, Jungseul Ok, and Sungsoo Ahn. Self-training large language models with confident reasoning. *arXiv preprint arXiv:2505.17454*, 2025.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International conference on machine learning*, pp. 1724–1732. PMLR, 2017.

Byungjai Kim, Chanho Ahn, Wissam J Baddar, Kikyung Kim, Huijin Lee, Saehyun Ahn, Seungju Han, Sungjoo Suh, and Eunho Yang. Test-time ensemble via linear mode connectivity: A path to better adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025.

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 *International Conference on Learning Representations*, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857, 2022.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, pp. 1–34, 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.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? In *Advances in Neural Information Processing Systems*, volume 34, pp. 21808–21820, 2021.

Robert A Marsden, Mario Döbler, and Bin Yang. Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In *Winter Conference on Applications of Computer Vision*, pp. 2555–2565, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PmLR, 2016.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: scaling for compute and sample efficient continuous control. *Advances in neural information processing systems*, 37:113038–113071, 2024.

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 *Internetional Conference on Learning Representations*, pp. 1–14, 2023.

Shuaicheng Niu, Chunyan Miao, Guohao Chen, Pengcheng Wu, and Peilin Zhao. Test-time model adaptation with only forward passes. In *International Conference on Machine Learning*, 2024.

Shuaicheng Niu, Guohao Chen, Peilin Zhao, Tianyi Wang, Pengcheng Wu, and Zhiqi Shen. Self-bootstrapping for versatile test-time adaptation. *arXiv preprint arXiv:2504.08010*, 2025.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *International conference on machine learning*, pp. 5628–5637. PMLR, 2019.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Max Schwarzer, Johan Samir Obando Ceron, Aaron Courville, Marc G Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, pp. 30365–30380. PMLR, 2023.

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, 2020.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. Mathscale: Scaling instruction tuning for mathematical reasoning. In *Forty-first International Conference on Machine Learning*, 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*, pp. 1–12, 2021.

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

Shoukai Xu, Mingkui Tan, Liu Liu, Zhong Zhang, Peilin Zhao, et al. Test-time adapted reinforcement learning with action entropy regularization. In *International Conference on Machine Learning*, 2025.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pp. 12310–12320. PMLR, 2021.

Chaoning Zhang, Kang Zhang, Chenshuang Zhang, Trung X Pham, Chang D Yoo, and In So Kweon. How does simsiam avoid collapse without negative samples? a unified understanding with self-supervised contrastive learning. In *International Conference on Learning Representations*, 2022.

Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. Come: Test-time adaption by conservatively minimizing entropy. In *International Conference on Learning Representations*, 2025.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

APPENDIX

CONTENTS

A Related Work

In this section, we connect test-time entropy minimization and self-supervised learning through their shared challenge of collapse, and relate our approach to these areas.

Test-Time Entropy minimization seeks to promote confident predictions by minimizing the prediction entropy on test data. Tent (Wang et al., 2021) first exploits this scheme for test-time model adaptation to novel and out-of-distribution environments. Unlike *test-time training* approaches (Sun et al., 2020; Liu et al., 2021; Bartler et al., 2022; Gandelsman et al., 2022), which train an extra auxiliary self-supervised branch to produce learning signals from test data, test-time entropy minimization (Wang et al., 2021; Niu et al., 2022; Marsden et al., 2024; Lee et al., 2024; Zhang et al., 2025; Hu et al., 2025b) removes reliance on the source training process and adapts an arbitrary model on-the-fly by using its own predicted labels for self-training, rendering it practical in real-world applications (Niu et al., 2024; Liang et al., 2025). Recently, test-time entropy optimization has attracted growing interest in various fields, such as incentivizing knowledge in large language models during inference (Jang et al., 2025; Zuo et al., 2025; Hu et al., 2025a), and adapting a fixed policy network to evolving environments in real-time (Xu et al., 2025), showing great potential.

Nevertheless, as highlighted by Niu et al. (2023), entropy minimization is inherently unstable and prone to collapse, *e.g.*, converging towards constant one-hot outputs that trivially minimize the entropy loss without meaningful predictions. To improve stability, subsequent methods explored strategies such as sample filtering (Niu et al., 2023; Lee et al., 2024; Hu et al., 2025a), uncertainty quantification (Zhang et al., 2025), and parameter restoration (Chen et al., 2024; Marsden et al., 2024). Despite these advances, collapsed constant solutions remain valid optima in prior methods, so stability often hinges on heuristics, thresholds, or extra compute, leaving methods sensitive to architectures and domains, where robust architectural design is largely underexplored. Our ZeroSiam fills this gap by introducing asymmetry that rules out trivial minima in test-time entropy minimization, while remaining efficient by avoiding augmentations and extra encoder passes.

Self-supervised Learning aims to learn discriminative and consistent representations by creating pretext tasks such as contrastive learning (Oord et al., 2018; Saunshi et al., 2019; He et al., 2020; Chen et al., 2020). A central challenge in SSL is trivial collapse, where all outputs degenerate to a constant to maximize representation similarity. Contrastive methods such as SimCLR (Chen et al., 2020) and MoCo (He et al., 2020) prevent collapse by pushing apart negative pairs, but rely on large batch sizes, memory banks, or careful negative mining strategies. To remove dependency on negative samples, BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021) introduce asymmetric architectures, preventing collapse with a *predictor–stop-gradient asymmetry* to break the stability of the collapsed constant solution. Although subsequent methods such as Barlow Twins (Zbontar et al., 2021) and VicReg (Bardes et al., 2022) achieve stability without symmetry-breaking design by maximizing feature diversity within a batch, they implicitly assume large batches and i.i.d. streams, which often fail during testing. Inspired by Simsiam, we revisit asymmetry in the context of single-branch, entropy-based TTA, and show how it can be instantiated in a minimal and efficient manner.

B THEORETICAL ANALYSIS

Notations. Let $\mathcal{L} = H(p^o) + \alpha\, D(p^o \,\|\, \mathrm{sg}[p^r])$ be the optimization objective of ZeroSiam, where $H(\cdot)$ is the entropy loss, $D(\cdot)$ the alignment regularizer, and $p^o, p^r \in \Delta^{|\mathcal{C}|-1}$ are the probability distributions induced by the online and target branches. Let the encoder f and classifier g be pretrained, and h a lightweight *linear* predictor. At initialization, we set h as the identity mapping $\mathbf{I}$, so that the online and target branches coincide and both lie in the non-collapse region $\{p \in \Delta^{|\mathcal{C}|-1} | \min_c p_c \geq \delta > 0\}$.

B.1 ASSUMPTIONS

Assumption 1. ***(Lipschitz continuity of the encoder)*** *Let $f(\cdot;\theta_f)$ be an encoder parameterized by θ_f. We assume f is Lipschitz continuous with respect to its parameters. Specifically, there exists a constant $L_f > 0$ such that for any θ_1, θ_2 and any input $\mathbf{x}$ in the domain,*

$$\|f_{\theta_1}(\mathbf{x}) - f_{\theta_2}(\mathbf{x})\|_2 \leq L_f \|\theta_1 - \theta_2\|.$$

Assumption 1 bounds the sensitivity of the encoder f to parameter perturbations, thereby ensuring stability for subsequent optimization analysis, and is commonly employed in prior theoretical studies on representation learning and model convergence.

Assumption 2. ***(Standard optimization assumptions)*** *Let $\mathcal{L}(\cdot)$ denote the objective function, $\phi = (\theta_h, \theta_f)$ the set of optimization parameters, and η_h, η_f the learning rates for the predictor and encoder, respectively. Our analysis relies on the following standard assumptions, which are common in the optimization literature:*
(1) Smoothness: $\mathcal{L}$ is ρ-smooth. Specifically, there exists $\rho > 0$ such that for all ϕ_1, ϕ_2,

$$\|\nabla_\phi \mathcal{L}(\phi_1) - \nabla_\phi \mathcal{L}(\phi_2)\| \leq \rho \|\phi_1 - \phi_2\|;$$

(2) Descent condition: The learning rates satisfy $\max(\eta_h, \eta_f) < \frac{1}{\rho}$, ensuring monotonic descent.

Assumption 2 constrains the appropriate range of learning rates to guarantee monotonic descent of $\mathcal{L}(\cdot)$, which is a common assumption in optimization research.

B.2 PROOF OF LEMMA 1

Lemma 1. ***(Gradient dominance in entropy descent)*** *Let $H(\cdot)$ denote the entropy function, $g(\cdot;\theta_g), h(\cdot;\theta_h)$ and $f(\cdot;\theta_f)$ denote the classifier, the predictor and the encoder, $\phi = (\theta_h, \theta_f)$ the set of optimization parameters, and η_h, η_f the learning rates for the predictor and encoder, respectively. Then the discrete change in entropy of the target branch satisfies:*

$$\|\nabla_\phi H(p^o(t))\| \geq \frac{1}{L_H \|g\|_2 L_f \cdot \max(\eta_h, \eta_f)} \cdot |H(p^r(t+1)) - H(p^r(t))|,$$

where L_H, L_f are the Lipschitz constants of $H(\cdot)$ and f, $\|g\|_2$ denotes the spectral norm of the classifier weight vector, and t is the optimization iteration step.

Proof. Consider the online and target branches:

$$\begin{cases} p^o = \mathrm{softmax}(g(h(f(\mathbf{x};\theta_f);\theta_h);\theta_g)) & \text{(online branch)}, \\ p^r = \mathrm{softmax}(g(f(\mathbf{x};\theta_f);\theta_g)) & \text{(target branch)}. \end{cases}$$

By design, the encoder f of the target branch is synchronized with the online branch after each update:

$$\theta_f^r(t+1) = \theta_f^o(t+1) = \theta_f^o(t) - \eta_f \nabla_{\theta_f} H(p^o(t)).$$

Thus, we bound the entropy change of the target branch $|H(p^r(t+1)) - H(p^r(t))|$ by tracing the propagation of parameter updates through the online branch.

According to the Assumption 1, the encoder f is L_f-Lipschitz continuous w.r.t. θ_f, we have

$$\|\mathbf{z}(t+1) - \mathbf{z}(t)\|_2 = \|f(\mathbf{x};\theta_f(t+1)) - f(\mathbf{x};\theta_f(t))\|_2 \leq L_f \|\theta_f(t+1) - \theta_f(t)\|. \tag{4}$$

where $\mathbf{z} = f(\mathbf{x}; \theta_f)$ is the output feature of the encoder.

Note that in the non-collapse region $\{p \in \Delta^{|\mathcal{C}|-1} | \min_c p_c \geq \delta > 0\}$, the entropy function $H(\cdot)$ is Lipschitz continuous with constant $L_H = \sqrt{|\mathcal{C}|}(|\log \delta| + 1)$, thus:

$$|H(p(t+1)) - H(p(t))| \leq L_H \|p(t+1) - p(t)\|_2. \tag{5}$$

Since the pre-trained classifier g is fixed, and the softmax function is 1-Lipschitz, combining Eqn. (4) and Eqn. (5), we have the following chain:

$$\begin{aligned} |H(p^r(t+1)) - H(p^r(t))| &\leq L_H \|p^r(t+1) - p^r(t)\|_2 \\ &\leq L_H \|g\|_2 \|\mathbf{z}^r(t+1) - \mathbf{z}^r(t)\|_2 \\ &\leq L_H \|g\|_2 L_f \|\theta_f^r(t+1) - \theta_f^r(t)\| \\ &= L_H \|g\|_2 L_f \eta_f \|\nabla_{\theta_f} H(p^o(t))\|, \end{aligned}$$

where $\|g\|_2$ is the spectral norm of the classifier weight vector.

Let $\phi = (\theta_h, \theta_f)$ denote the set of optimization parameters, and η_h, η_f the learning rates for the predictor and encoder. Since:

$$\eta_f \leq \max(\eta_h, \eta_f),$$
$$\|\nabla_{\theta_f} H(p^o(t))\| \leq \left\| \begin{bmatrix} \nabla_{\theta_f} H(p^o(t)) \\ \nabla_{\theta_h} H(p^o(t)) \end{bmatrix} \right\| = \|\nabla_\phi H(p^o(t))\|,$$

we obtain:

$$\|\nabla_\phi H(p^o(t))\| \geq \frac{1}{L_H \|g\|_2 L_f \cdot \max(\eta_h, \eta_f)} \cdot |H(p^r(t+1)) - H(p^r(t))|.$$

□

B.3 Proof of Lemma 2

Lemma 2. ***(Collapse Direction with Maximal Negative Curvature)*** *Let $H(\cdot)$ denote the entropy function, $u = g(h(f(\mathbf{x}; \theta_f); \theta_h); \theta_g)$ the uncertainty logits, and $p = softmax(u)$ the induced probability distribution. For any class $c \in \mathcal{C}$, there exists a direction $\mathbf{v} = (\mathbf{v}_h, \mathbf{v}_f) \in \mathbb{R}^{\dim(\phi)}$ such that:*
(1) p_c increases monotonically along $\mathbf{v}$;
(2) As $p_c \to 1$, it holds that $\mathbf{v}^\top \nabla_\phi^2 H \mathbf{v} \to \lambda_{\min}(\nabla_\phi^2 H)$;
(3) For any $\mathbf{w} \perp \mathbf{v}$, we have $\mathbf{v}^\top \nabla_\phi^2 H \mathbf{v} \leq \mathbf{w}^\top \nabla_\phi^2 H \mathbf{w}$.

Proof. (1) Let $\mathbf{z} = f(\mathbf{x}; \theta_f) \in \mathbb{R}^d$ be the encoder output, $u = g(h(\mathbf{z}; \theta_h); \theta_g) \in \mathbb{R}^{|\mathcal{C}|}$ the uncertainty logits, and $\mathbf{W}, \mathbf{P}$ the weight vector of g, h, respectively. Define $\mathbf{v} = (\mathbf{v}_h, \mathbf{v}_f)$ such that:

$$\begin{cases} \mathbf{v}_h = a \cdot \mathbf{W}_k^\top r, \\ \mathbf{v}_f = \nabla_{\theta_f}(r^\top \mathbf{z}); \end{cases}$$

where r is a feature direction satisfying $r^\top \mathbf{z} \neq 0$ and $a > 0$. $\mathbf{v}_f$ is the direction in normalization layer parameters that amplifies the feature component along r.

This joint direction $\mathbf{v} = (\mathbf{v}_h, \mathbf{v}_f)$ induces a change in logits:

$$\Delta u = \mathbf{W}(\mathbf{v}_h)\mathbf{z} + \mathbf{W}\mathbf{P}(\nabla_{\theta_f} \mathbf{z} \mathbf{v}_f) \approx a(r^\top \mathbf{z})\mathbf{W}\mathbf{W}_k^\top + \mathbf{W}\mathbf{P}r.$$

By choosing $r^\top \mathbf{z} > 0$ and sufficiently large a, we ensure:

$$[\Delta u]_c > 0, \quad [\Delta u]_j = \mathcal{O}(1) \text{ for } j \neq c.$$

Thus, moving along $\mathbf{v}$ increases u_c, and hence $p_c = \text{softmax}(u_c)$, monotonically.

(2) The Hessian of entropy $H(\cdot)$ w.r.t. logits u is:

$$[\nabla_u^2 H]_{ij} = \frac{\partial^2 H}{\partial u_i \partial u_j} = p_i(\delta_{ij} - p_j)(H - \log p_i - 1).$$

As $p_c \to 1$, let $\mu = 1 - p_c \to 0^+$, and for $p_j \approx \frac{\mu}{|\mathcal{C}|-1}$ for $j \neq c$. Then in the Hessian matrix

$$\begin{cases} \frac{\partial^2 H}{\partial u_c^2} = p_c(1-p_c)(H - \log p_c - 1), & \text{Diagonal for } c, \\ \frac{\partial^2 H}{\partial u_j^2} = p_j(1-p_j)(H - \log p_j - 1), & \text{Diagonal for } j \neq c, \\ \frac{\partial^2 H}{\partial u_i \partial u_j} = -p_i p_j(H - \log p_i - \log p_j - 1), & \text{Off-diagonal}, \end{cases}$$

we obtain the following approximation:

$$\begin{cases} \frac{\partial^2 H}{\partial u_c^2} \approx (1-\mu)\mu(H + \mu - 1) \approx \mu(\mu \log \frac{|\mathcal{C}|-1}{\mu} + \mu - 1) \approx -\mu, \\ \frac{\partial^2 H}{\partial u_j^2} \approx \frac{\mu}{|\mathcal{C}|-1}(\mu \log \frac{|\mathcal{C}|-1}{\mu} - \log \frac{\mu}{|\mathcal{C}|-1} - 1) \approx \frac{\mu}{|\mathcal{C}|-1} \log \frac{|\mathcal{C}|-1}{\mu} > 0, \\ \frac{\partial^2 H}{\partial u_i \partial u_j} \approx -p_i p_j(-\log p_j) > 0. \end{cases}$$

That is, as $p_c \to 1$, we have

$$\mathbf{v}^\top \nabla_\phi^2 H \mathbf{v} \to \lambda_{\min}(\nabla_\phi^2 H).$$

(3) The parameter space Hessian quadratic form along $\mathbf{v}$ is:

$$\mathbf{v}^\top \nabla_\phi^2 H \mathbf{v} = \Delta u^\top \nabla_u^2 H \Delta u + \sum_i \frac{\partial H}{\partial u_i} \mathbf{v}^\top \nabla_\phi^2 u_i \mathbf{v}.$$

As $p_c \to 1$, $\nabla_u H \to 0$, so the higher-order terms vanishes. Since $\Delta u = \frac{\partial u}{\partial \phi} \propto e_c$, we have:

$$\mathbf{v}^\top \nabla_\phi^2 H \mathbf{v} \propto e_c^\top \nabla_u^2 H e_c = \frac{\partial^2 H}{\partial u_c^2} < 0.$$

For any $\mathbf{w} \perp \mathbf{v}$, $\Delta u_{\mathbf{w}} = \frac{\partial u}{\partial \phi}\mathbf{w}$ is not parallel to e_c, so:

$$\mathbf{w}^\top \nabla_\phi^2 H \mathbf{w} \geq \mathbf{v}^\top \nabla_\phi^2 H \mathbf{v}.$$

□

B.4 Proof of Theorem 1

Theorem 1. ***(Optimization and Stability of ZeroSiam)*** *Consider the ZeroSiam objective $\mathcal{L} = H(p^o) + \alpha\, D(p^o \,\|\, \mathrm{sg}[p^r])$, where $H(\cdot)$ denotes the entropy loss, $D(\cdot)$ the alignment regularizer, and $p^o, p^r \in \Delta^{|\mathcal{C}|-1}$ are the probability distributions induced by the online and target branches. Given the encoder f, classifier g and predictor h. Under Assumptions 1 and 2, the following hold:*
(1) For $\alpha = 0$, the entropy variation satisfies $|\Delta H(p^o)| > |\Delta H(p^r)|$, and the Hessian of $H(p^o)$ attains its minimal eigenvalue along collapse directions $\mathbf{v}$:

$$\lambda_{\min}\big(\nabla^2 H(p^o)\big) = \mathbf{v}^\top \nabla^2 H(p^o) \mathbf{v}.$$

(2) For $\alpha > 0$, the predictor h serves as a filtering mechanism that suppresses gradient update directions corresponding to over-amplified logits, and the system converges to a stable equilibrium: there exists $h_{\min} > 0$ such that

$$H(p^o) > h_{\min},\ p^o \to p^r.$$

Proof. (1) By Lemma 1, the target branch entropy variation is determined by the gradient of the online branch:

$$\|\nabla_\phi H(p^o(t))\| \geq \frac{1}{L \cdot \eta_{\max}} \cdot |H(p^r(t+1)) - H(p^r(t))|, \tag{6}$$

where $L = L_H \|g\|_2 L_f$ and $\eta_{\max} = \max(\eta_h, \eta_f)$.

Under the Assumption 2 that $\mathcal{L} = H(\cdot)$ is ρ-smoothness and $\eta_{\max} < \frac{1}{\rho}$, the descent lemma gives:

$$H(p^o(t+1)) \leq H(p^o(t)) - \nabla_\phi H^\top D_\eta \nabla_\phi H + \frac{\rho}{2} \|D_\eta \nabla_\phi H\|^2,$$

where $D_\eta = \mathrm{diag}(\eta_f \mathbf{I}_{d_f}, \eta_h \mathbf{I}_{d_h})$. So we have:

$$|\Delta H^o| = H(p^o(t+1)) - H(p^o(t)) \geq \eta_{\min} \|\nabla_\phi H\|^2,\ \eta_{\min} = \min(\eta_h, \eta_f). \tag{7}$$

Combining Eqn. (6) and Eqn. (7), we obtain:

$$\frac{|\Delta H^o|}{|\Delta H^r|} \geq \frac{\eta_{\min}\|\nabla_\phi H\|^2}{L \cdot \eta_{\max}\|\nabla_\phi H\|} = \frac{\eta_{\min}}{\eta_{\max}} \cdot \frac{\|\nabla_\phi H\|}{L}.$$

Since $L, \|\nabla_\phi H\| > 0$ and bounded in non-collapse regions, achieving $\frac{|\Delta H^o|}{|\Delta H^r|} > 1$ requires:

$$\frac{\eta_{\min}}{\eta_{\max}} > \frac{L}{\|\nabla_\phi H\|}.$$

In practice, we set $\eta_h = k \cdot \eta_f$ with $k > 1$, so there always exists a k such that

$$|\Delta H(p^o)| > |\Delta H(p^r)|.$$

We further construct the direction $\mathbf{v} = (\mathbf{v}_h, \mathbf{v}_f) \in \mathbb{R}^{\dim(\phi)}$ according to Lemma 2, the corresponding directional derivative is given by:

$$D_{\mathbf{v}} H = \nabla_\phi H^\top \mathbf{v} \propto [\nabla_u H]_c,$$

where u is the uncertainty logits. And the second-order directional derivative is:

$$D^2_{\mathbf{v}} H = \mathbf{v}^\top \nabla^2_\phi H \mathbf{v} = \lambda_{\min} < 0.$$

This implies that along $\mathbf{v}$, the loss decreases fastest at second order (Jin et al., 2017).

Moreover, by Lemma 2, p_c increases monotonically along $\mathbf{v}$, thus $[\nabla_u H]_c = p_c(H - \log p_c - 1)$ increases synchronously, creating a self-reinforcing feedback loop that accelerates collapse.

(2) Consider the optimization objective:

$$\mathcal{L} = H(p^o) + \alpha\, D(p^o \,\|\, \mathrm{sg}[p^r]),$$

where $D(p^o \,\|\, \mathrm{sg}[p^r]) = D_{\mathrm{KL}}(p^o \,\|\, \mathrm{sg}[p^r]) + D_{\mathrm{KL}}(\mathrm{sg}[p^r] \,\|\, p^o)$ are symmetric KL loss.

Under the stop-gradient setting on the target branch, the gradient of the shared encoder f is solely determined by the online branch. We have:

$$\nabla_{\mathbf{z}} \mathcal{L} = (\frac{\partial u^o}{\partial \mathbf{z}}) \nabla_{u^o} \mathcal{L} = \mathbf{P}^\top \mathbf{W}^\top [\nabla_{u^o} H(p^o) + 2\alpha(p^o - p^r)],$$

where $\mathbf{z} = f(\mathbf{x}; \theta_f)$ is the encoder output, $\mathbf{W}, \mathbf{P}$ the weight vector of the classifier g and the predictor h, respectively. Consider a parameter update unit direction $\mathbf{v_h}$, $\|\mathbf{v_h}\|_2 = 1$, then:

$$< \nabla_{\mathbf{z}} \mathcal{L}, \mathbf{v_h} >= \mathbf{v_h}^\top \nabla_{\mathbf{z}} \mathcal{L} = \mathbf{v_h}^\top (\frac{\partial u^o}{\partial \mathbf{z}}) \nabla_{u^o} \mathcal{L} = \mathbf{P}^\top \mathbf{W}^\top [\nabla_{u^o} H(p^o) + 2\alpha(p^o - p^r)]$$

During the minimization of $\mathcal{L}$, the predictor h encourages the update direction that ensuring

$$p^o \to p^r,$$

and suppresses gradient update directions corresponding to over-amplified logits.

We further analyze the convergence properties of the optimization system.

By the definition of entropy and KL divergence, we have:

$$H(p^o) = -\sum_{c=1}^{|\mathcal{C}|} p_c^o \log p_c^o = -\sum_{c=1}^{|\mathcal{C}|} p_c^o \log p_c^r - D_{\mathrm{KL}}(p^o || p^r).$$

Applying Gibbs' inequality, $-\sum_{c=1}^{|\mathcal{C}|} p_c^o \log p_c^r \geq H(p^r)$, we obtain:

$$H(p^o) \geq H(p^r) - D_{\mathrm{KL}}(p^o || p^r).$$

Since $D_{\mathrm{SKL}}(p^o || p^r) \geq D_{\mathrm{KL}}(p^o || p^r)$, it follows that:

$$H(p^o) \geq H(p^r) - D_{\mathrm{SKL}}(p^o || p^r). \tag{8}$$

When the target branch is still in the non-collapse region $\{p^r \in \Delta^{|\mathcal{C}|-1} | \min_c p_c^r \geq \delta > 0\}$, the entropy satisfies:

$$H(p^r) \geq -\log(1-(1-|\mathcal{C}|)\delta) > 0. \tag{9}$$

Define the Lyapunov function $V(t) = \mathcal{L}(t)$. Since $H(p^o) \geq 0$ and $D_{\text{SKL}}(p^o||p^r) \geq 0$, we have

$$V(t) > 0.$$

Moreover, under the Assumption 2, the descent lemma guarantees:

$$V(t+1) \geq V(t) - \frac{\eta_{\min}}{2}\|\nabla_\phi \mathcal{L}(t)\|^2 \leq V(t).$$

Thus, $V(t)$ is monotonically decreasing and bounded below. According to the Monotone Convergence Theorem, $V(t) \to V^* > 0$. In particular:

$$D_{\text{SKL}}(p^o(t)||p^r(t)) \leq \frac{V(0)}{\alpha}. \tag{10}$$

Combining Eqn. (8), Eqn. (9), and Eqn. (10):

$$H(p^o) \geq H(p^r) - D_{\text{SKL}}(p^o||p^r) \geq -\log(1-(1-|\mathcal{C}|)\delta) - \frac{V(0)}{\alpha}.$$

Define

$$h_{\min} = -\log(1-(1-|\mathcal{C}|)\delta) - \frac{V(0)}{\alpha}.$$

Let the hyperparameter $\alpha > \frac{V(0)}{-\log(1-(1-|\mathcal{C}|)\delta)}$, then $h_{\min} > 0$.

Overall, we have proven that the predictor h serves as a filtering mechanism that suppresses gradient update directions corresponding to over-amplified logits, and the system converges to a stable equilibrium: there exists $h_{\min} > 0$ such that

$$H(p^o) > h_{\min},\ p^o \to p^r.$$

□

Remark. Together, the theoretical analyses for both $\alpha = 0$ and $\alpha > 0$ give complementary insights into how ZeroSiam introduces meaningful discrepancies for regularization and bias filtration, while also demonstrating its convergence toward a stable, and non-collapsing equilibrium, which aligns with our empirical findings of ZeroSiam's stability across diverse challenging adaptation scenarios.

C More Implementation Details

C.1 More Details on Dataset

For vision adaptation, we evaluate the out-of-distribution generalization ability of all methods on a large-scale and widely used benchmark, namely **ImageNet-C**[1] (Hendrycks & Dietterich, 2019). ImageNet-C is constructed by corrupting the original ImageNet (Deng et al., 2009) test set. The corruption (as shown in Figure A) consists of 15 different types, *i.e.*, Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transformation, pixelation, and JPEG compression, where each corruption type has 5 severity levels and the larger severity level means more severe distribution shift.

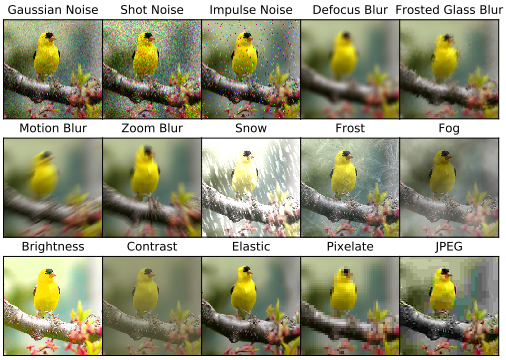

Figure A: Visualizations of different corruption types in ImageNet-C benchmark.

For natural language reasoning, we evaluate the effectiveness of online reasoning incentivization on a comprehensive mathematical reasoning benchmark, including Math-500 (Lightman et al., 2023), CollegeMath (Tang et al., 2024), AIME24 (Codeforces, 2024), and Minerva (Lewkowycz et al., 2022). Specifically, Math-500 is derived from the larger MATH benchmark (Hendrycks et al., 2021), consisting of difficult mathematics problems originally taken from high school competitions. It spans domains such as pre-algebra, algebra, number theory, and calculus, emphasizing abstract reasoning and multi-step problem solving. CollegeMath is a college-level mathematics dataset constructed from nine open-access college textbooks, covering seven core areas: algebra, pre-calculus, calculus, vector calculus, probability, linear algebra, and differential equations. The full dataset contains 2818 test questions, from which we sample 1200 for efficient evaluation. AIME24 consists of 30 advanced problems from the 2024 American Invitational Mathematics Examination (AIME) I and II, designed to test extended chain-of-thought reasoning. Minerva includes 272 undergraduate-level mathematics and science questions from MIT OpenCourseWare (OCW), aimed at assessing models' reasoning ability in scientific problem-solving contexts.

C.2 More Experimental Protocols on Methods

All pre-trained models involved in our paper are publicly available, including ResNet50-GN[2], ViT-Base[3], ViT-Small[4], ConvNeXt-Tiny[5] and Swin-Tiny[6] obtained from `timm` repository (Wightman,

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

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

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

[4] https://storage.googleapis.com/vit_models/augreg/S_16-i21k-300ep-lr_0.001-aug_light1-wd_0.03-do_0.0-sd_0.0--imagenet2012-steps_20k-lr_0.03-res_224.npz

[5] https://github.com/rwightman/pytorch-image-models/releases/download/v0.1-rsb-weights/convnext_tiny_hnf_a2h-ab7e9df2.pth

[6] https://github.com/SwinTransformer/storage/releases/download/v1.0.0/swin_tiny_patch4_window7_224.pth

2019) for image classification, and Llama3.1-8B-Instruct[7] (Dubey et al., 2024) for natural language reasoning. In the following, we provide the implementation details of our proposed method and all comparative methods evaluated in our experiments, which help reproduce our results.

ZeroSiam (Ours). We use symmetric KL as the divergence term in Eqn. (3), and α is fixed to 1 without tuning. For vision adaptation, we use SGD as the update rule, with a momentum of 0.9, batch size of 64 (except for the experiments of batch size = 1), learning rate η_f of 0.0025 / 0.005 / 0.001 / 0.00025 / 0.0005 with learning rate η_h of 0.025 / 0.005 / 0.01 / 0.0025 / 0.0025 for ResNet50-GN / ViT-Base / ViT-Small / ConvNeXt-Tiny / Swin-Tiny, respectively. The learning rate for batch size = 1 is scaled down by 16 for ResNet50-GN and 32 for other models following SAR. The trainable parameters are all the affine parameters of normalization layers. For natural language reasoning, following the configuration for TLM, we use AdamW as the update rule, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.0, and optimize the prediction entropy of the first 8 output tokens. Both learning rates η_f and η_h are set to 7.5×10^{-6} / 5×10^{-6} / 1.25×10^{-5} / 5×10^{-6} for Math-500 / CollegeMath / AIME24 / Minerva. Trainable parameters are the LoRA parameters with a rank of 8.

Tent[8] (Wang et al., 2021). We follow all hyper-parameters that are set in Tent unless it does not provide. Specifically, for vision adaptation, we use SGD as the update rule, with a momentum of 0.9, batch size of 64 (except for the experiments of batch size = 1), and learning rate of 0.00025 / 0.001 / 0.0001 / 0.000025 / 0.00005 for ResNet50-GN / ViT-Base / ViT-Small / ConvNeXt-Tiny / Swin-Tiny, respectively. The learning rate for batch size = 1 is scaled down to (0.00025/32) for ResNet50-GN, (0.001/64) for ViT-Base, (0.0001/64) for ViT-Small, (0.000025/64) for ConvNeXt-Tiny and (0.00005/64) for Swin-Tiny. The trainable parameters are all the affine parameters of normalization layers. For natural language reasoning, we use AdamW as the update rule, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.0, and optimize the prediction entropy of the first 8 output tokens. The learning rate is 7.5×10^{-6} / 5×10^{-6} / 1.25×10^{-5} / 5×10^{-6} for Math-500 / CollegeMath / AIME24 / Minerva. The trainable parameters are the LoRA parameters with a rank of 8.

SAR[9] (Niu et al., 2023). We follow all hyper-parameters that are set in SAR unless it does not provide. Specifically, for vision adaptation, we use SGD as the update rule, with a momentum of 0.9, batch size of 64 (except for the experiments of batch size = 1), and learning rate of 0.00025 / 0.001 / 0.0001 / 0.000025 / 0.00005 for ResNet50-GN / ViT-Base / ViT-Small / ConvNeXt-Tiny / Swin-Tiny. The learning rate for batch size = 1 is scaled down to (0.00025/16) for ResNet50-GN, (0.001/32) for ViT-Base, (0.0001/32) for ViT-Small, (0.000025/32) for ConvNeXt-Tiny and (0.00005/32) for Swin-Tiny. The entropy threshold E_0 is set to $0.4\times \ln C$, where C is the number of task classes. The trainable parameters are the affine parameters of norm layers from layer 1 to layer 3 in ResNet50-GN, from blocks 1 to blocks 8 in ViT-Base and ViT-Small, and all affine parameters in other models. For natural language reasoning, E_0 is set to 0.4. We use AdamW as the update rule, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.0, and optimize the prediction entropy of the first 8 output tokens. The learning rate is 7.5×10^{-6} / 5×10^{-6} / 1.25×10^{-5} / 5×10^{-6} for Math-500 / CollegeMath / AIME24 / Minerva, trainable parameters are the LoRA parameters with a rank of 8.

EATA[10] (Niu et al., 2022). We follow all hyper-parameters that are set in EATA unless it does not provide. Specifically, for vision adaptation, the entropy constant E_0 (for reliable sample identification) is set to $0.4\times \ln C$, where C is the number of task classes. The ϵ for redundant sample identification is set to 0.05. The trade-off parameter β for entropy loss and regularization loss is set to 2,000. The number of pre-collected in-distribution test samples for Fisher importance calculation is 2,000. The update rule is SGD, with a momentum of 0.9, batch size of 64 (except for the experiments of batch size = 1), and learning rate of 0.00025 / 0.001 / 0.0001 / 0.000025 / 0.00005 for ResNet50-GN / ViT-Base / ViT-Small / ConvNeXt-Tiny / Swin-Tiny. The learning rate for batch size = 1 is scaled down to (0.00025/32) for ResNet50-GN, (0.001/64) for ViT-Base, (0.0001/64) for ViT-Small, (0.000025/64) for ConvNeXt-Tiny and (0.00005/64) for Swin-Tiny. The trainable parameters are all affine parameters of normalization layers. For natural language reasoning, E_0 is set to 0.4, while the anti-forgetting regularizer is not applied due to a lack of knowledge for in-distribution data that the large language model trains on. We use AdamW as the update rule, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.0, and optimize the prediction entropy of the first 8 output tokens. The

[7] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[8] https://github.com/DequanWang/tent
[9] https://github.com/mr-eggplant/SAR
[10] https://github.com/mr-eggplant/EATA

learning rate is 7.5×10^{-6} / 5×10^{-6} / 1.25×10^{-5} / 5×10^{-6} for Math-500 / CollegeMath / AIME24 / Minerva. The trainable parameters are the LoRA parameters with a rank of 8.

DeYO[11] **(Lee et al., 2024).** We follow all hyper-parameters that are set in DeYO unless it does not provide. Specifically, the entropy constant E_0 (for reliable sample identification) is set to $0.4 \times \ln 1000$, and the factor τ_{Ent} is set to $0.5\times \ln 1000$. The Pseudo-Label Probability Difference (PLPD) threshold τ_{PLPD} is set to 0.2. The update rule is SGD, with a momentum of 0.9, batch size of 64 (except for the experiments of batch size = 1), and learning rate of 0.00025 / 0.001 / 0.0001 / 0.000025 / 0.00005 for ResNet50-GN / ViT-Base / ViT-Small / ConvNeXt-Tiny / Swin-Tiny. The learning rate for batch size = 1 is scaled down to (0.00025/16) for ResNet50-GN, (0.001/32) for ViT-Base, (0.0001/32) for ViT-Small, (0.000025/32) for ConvNeXt-Tiny and (0.00005/32) for Swin-Tiny. Trainable parameters are the affine parameters of norm layers from layer 1 to 3 in ResNet50-GN, from blocks 1 to 8 in ViT-Base and ViT-Small, and all affine parameters in other models.

COME[12] **(Zhang et al., 2025).** For vision adaptation, we implement COME based on DeYO for comparisons, following the hyper-parameters set in DeYO unless otherwise specified. Specifically, the entropy constant E_0 (for reliable sample identification) is set to $0.4 \times \ln 1000$, and the factor τ_{Ent} is set to $0.5\times \ln 1000$. The Pseudo-Label Probability Difference (PLPD) threshold τ_{PLPD} is set to 0.2. The uncertainty mass is set to C, where C is the number of task classes. The update rule is SGD, with a momentum of 0.9, batch size of 64 (except for the experiments of batch size = 1), and learning rate of 0.00025 / 0.001 / 0.0001 / 0.000025 / 0.00005 for ResNet50-GN / ViT-Base / ViT-Small / ConvNeXt-Tiny / Swin-Tiny. The learning rate for batch size = 1 is scaled down to (0.00025/16) for ResNet50-GN, (0.001/32) for ViT-Base, (0.0001/32) for ViT-Small, (0.000025/32) for ConvNeXt-Tiny and (0.00005/32) for Swin-Tiny. The trainable parameters are the affine parameters of norm layers from layer 1 to layer 3 in ResNet50-GN, from blocks 1 to blocks 8 in ViT-Base and ViT-Small, and all affine parameters in other models. For natural language reasoning, COME is implemented based on Tent, and uncertainty mass is set to $|Z|$, where $|Z|$ denotes the vocabulary size. We use AdamW as the update rule, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.0, and optimize the prediction entropy of the first 8 output tokens. The learning rate is 7.5×10^{-6} / 5×10^{-6} / 1.25×10^{-5} / 5×10^{-6} for Math-500 / CollegeMath / AIME24 / Minerva. The trainable parameters are the LoRA parameters with a rank of 8.

TLM (Hu et al., 2025a)[13]**.** We follow all hyper-parameters that are set in TLM unless it does not provide. Specifically, we compare with TLM on the natural language reasoning task, the perplexity threshold $\mathcal{P}$, is set to e^3. The update rule is AdamW, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay = 0.0. The learning rate is 7.5×10^{-6} / 5×10^{-6} / 1.25×10^{-5} / 5×10^{-6} for Math-500 / CollegeMath / AIME24 / Minerva. The trainable parameters are the LoRA parameters with a rank of 8.

[11] https://github.com/Jhyun17/DeYO
[12] https://github.com/BlueWhaleLab/COME
[13] https://github.com/Fhujinwu/TLM

D MORE EXPERIMENTAL RESULTS

In this section, we provide the detailed results of Tables 3-2 in the main paper and include additional experiments under both mild and wild testing scenarios to enable a comprehensive comparison.

D.1 DETAILED AND ADDITIONAL RESULTS UNDER WILD TEST SCENARIOS

We first provide the detailed results of TTA under label shifts and a single sample. From Tables A & B, ZeroSiam achieves superior performance across nearly all cases, *e.g.*, 51.6% (Ours) *vs.* 43.9% (DeYO) on ResNet50-GN under label shifts, and 51.3% (Ours) *vs.* 42.3% (COME) on ViT-Small under the batch size of 1. Similar results are observed even when the data stream exhibits both label shifts and data scarcity, as shown in Table C, suggesting our effectiveness across test settings.

D.2 DETAILED RESULTS FOR TTA ON THE BLIND-SPOT SUBSET

From Table D, prior methods tend to collapse when adapting on the blind-spot subset, *e.g.*, 18.9% (DeYO) *vs.* 30.6% (NoAdapt) w.r.t. the average accuracy on R50-GN. In contrast, even under such a challenging setting, ZeroSiam achieves consistent improvement across all domains and models, suggesting that ZeroSiam substantially expands the scope and reliability of TTA in the real world.

D.3 DETAILED RESULTS UNDER THE MILD TEST SCENARIO

ZeroSiam is not only effective under challenging test scenarios, but also enhances TTA performance significantly on the mild test scenario (Wang et al., 2021), where data comes with shuffled labels. As shown in Table E, ZeroSiam demonstrates superior performance on 4 out of 5 models and also achieves comparative results on ViT-Base, *e.g.*, the average accuracy of 48.6% (Ours) *vs.* 41.7% (DeYO) on ResNet50-GN, and 62.6% (Ours) *vs.* 62.8% (DeYO) on ViT-Base. This suggests that our ZeroSiam's design helps improve the stability and efficacy of TTA across a wide range of scenarios.

Table A: Detailed results of TTA under **IMBALANCED LABEL SHIFTS**, Table 3 in the main paper.

	Noise			Blur				Weather				Digital				
Model+Method	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet50-GN	18.0	19.8	17.9	19.8	11.4	21.4	24.9	40.4	47.3	33.6	69.3	36.3	18.6	28.4	52.3	30.6
• Tent	2.6	3.3	2.7	13.9	7.9	19.5	17.0	16.5	21.9	1.8	70.5	42.2	6.6	49.4	53.7	22.0
• SAR	33.1	36.5	35.5	19.2	19.5	33.3	27.7	23.9	45.3	50.1	71.9	46.7	7.1	52.1	56.3	37.2
• EATA	27.0	28.3	28.1	14.9	17.1	24.4	25.3	32.2	32.0	39.8	66.7	33.6	24.5	41.9	38.4	31.6
• COME	17.7	19.7	17.7	19.7	11.2	21.2	24.8	40.2	47.8	33.7	68.9	35.6	18.6	27.7	51.8	30.4
• DeYO	42.5	44.9	43.8	22.2	16.3	41.0	13.2	52.2	51.5	39.7	73.4	52.6	46.9	59.3	59.3	43.9
• ZeroSiam (ours)	42.9	45.1	43.1	35.1	36.1	44.5	49.2	58.1	55.1	62.9	72.3	54.8	52.5	61.7	60.1	**51.6**
ViT-Base	9.4	6.7	8.3	29.1	23.4	34.0	27.0	15.8	26.3	47.4	54.7	43.9	30.5	44.5	47.6	29.9
• Tent	32.7	1.4	34.6	54.4	52.3	58.2	52.2	7.7	12.0	69.3	76.1	66.1	56.7	69.4	66.4	47.3
• SAR	46.5	43.1	48.9	55.3	54.3	58.9	54.8	53.6	46.2	69.7	76.2	66.2	60.9	69.6	66.6	58.0
• EATA	35.9	34.6	36.7	45.3	47.2	49.3	47.7	56.5	55.4	62.2	72.2	21.7	56.2	64.7	63.7	50.0
• COME	45.3	52.1	52.0	56.1	57.4	62.5	60.7	66.6	64.0	71.6	77.1	65.9	61.9	72.9	69.2	62.4
• DeYO	53.5	36.0	54.6	57.6	58.7	63.7	46.2	67.6	66.0	73.2	77.9	66.7	69.0	73.5	70.3	62.3
• ZeroSiam (ours)	52.3	52.6	53.4	57.7	58.7	62.7	61.1	67.6	66.0	73.3	78.0	67.0	67.9	73.5	70.1	**64.1**
ViT-Small	2.0	2.0	1.5	24.2	17.1	30.3	22.0	9.5	19.2	37.9	44.4	30.1	24.8	38.2	41.0	22.9
• Tent	0.3	0.4	0.2	43.1	37.0	45.4	36.3	5.5	24.2	58.4	65.8	54.6	26.6	56.6	54.9	34.0
• SAR	1.6	2.0	1.3	44.3	39.0	46.5	39.0	16.6	45.9	58.6	65.8	54.7	44.4	56.9	55.2	38.1
• EATA	15.6	15.5	17.8	42.4	40.3	44.8	41.6	46.2	48.7	58.9	65.3	52.9	50.5	57.2	56.2	43.6
• COME	0.1	0.2	0.1	47.4	47.2	52.8	10.5	35.5	53.8	63.9	70.7	58.1	56.7	63.4	60.2	41.4
• DeYO	0.1	0.2	0.1	48.4	47.7	53.4	16.2	47.0	54.6	65.0	71.0	59.3	58.5	64.1	61.0	43.1
• ZeroSiam (ours)	25.7	25.9	27.9	48.2	48.3	53.5	51.5	55.7	55.6	65.4	71.2	58.4	59.5	64.2	61.2	**51.5**
ConvNeXt-Tiny	21.4	27.7	22.1	24.5	11.0	32.5	31.2	44.5	52.5	39.5	72.0	44.8	23.1	17.7	55.8	34.7
• Tent	18.1	23.8	26.6	24.3	4.2	33.9	29.2	41.6	46.3	39.7	73.2	51.6	18.4	10.7	56.9	33.2
• SAR	34.8	36.7	34.9	27.1	4.1	35.0	30.8	44.3	47.6	8.4	73.1	51.1	19.4	24.7	56.7	35.2
• EATA	34.4	36.8	35.0	26.7	19.1	37.0	33.3	47.7	48.4	50.6	73.0	51.9	28.7	31.4	56.9	40.7
• COME	42.6	44.5	42.4	38.7	1.2	46.0	11.8	57.1	56.9	61.8	75.6	60.6	5.3	51.9	61.9	43.9
• DeYO	33.2	38.0	36.7	27.9	2.6	36.4	9.2	52.7	45.3	2.9	74.4	56.6	4.7	19.5	58.4	33.2
• ZeroSiam (ours)	42.3	44.5	42.1	37.4	33.5	45.3	44.0	56.5	55.4	62.9	74.7	59.4	44.3	51.7	60.0	**50.3**
Swin-Tiny	26.6	27.9	22.8	21.7	13.2	26.5	25.4	41.0	45.8	47.7	67.6	39.1	22.6	8.4	33.3	31.3
• Tent	16.1	14.1	10.5	17.1	16.9	23.5	20.0	30.4	28.3	11.3	69.7	47.8	9.3	1.1	43.2	24.0
• SAR	29.8	28.0	29.8	19.8	10.6	24.4	21.2	33.1	34.3	22.3	67.9	45.9	13.2	4.8	42.9	28.5
• EATA	36.2	37.1	33.6	26.3	29.8	38.2	37.3	47.3	44.5	51.4	70.3	42.4	40.9	34.8	46.5	41.1
• COME	41.8	43.6	43.2	36.5	9.8	46.7	2.4	54.6	52.3	60.0	72.0	47.4	4.3	3.1	55.8	38.2
• DeYO	40.1	43.3	42.4	20.9	32.0	42.2	12.3	49.4	49.0	55.7	72.3	53.3	38.5	42.1	53.3	43.1
• ZeroSiam (ours)	44.4	45.8	45.7	36.4	38.3	47.0	47.0	54.3	51.9	58.7	72.4	54.2	52.7	51.8	55.5	**50.4**

Table B: Detailed results of TTA with **BATCH SIZE=1**.

	Noise			Blur				Weather				Digital				
Model+Method	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet50-GN	18.0	19.8	17.9	19.8	11.4	21.4	24.9	40.4	47.3	33.6	69.3	36.3	18.6	28.4	52.3	30.6
• Tent	2.5	2.9	2.5	13.5	3.6	18.6	17.6	15.3	23.0	1.4	70.4	42.2	6.2	49.2	53.8	21.5
• SAR	23.4	26.6	23.0	18.4	15.4	28.6	30.4	44.9	44.7	25.7	72.3	44.5	14.8	47.0	56.1	34.5
• EATA	24.8	28.3	25.7	18.1	17.3	28.5	29.3	44.5	44.3	41.6	70.9	44.6	27.0	46.8	55.7	36.5
• COME	17.8	19.7	17.7	19.7	11.3	21.4	24.9	40.3	47.6	33.6	68.9	35.7	18.5	27.9	51.8	30.5
• DeYO	41.8	44.7	43.0	22.5	24.7	41.8	24.4	54.5	52.2	20.7	73.5	53.5	48.5	60.2	59.8	44.4
• ZeroSiam (ours)	41.7	44.4	41.8	32.8	35.8	45.2	49.5	58.4	55.5	63.8	73.7	54.8	55.1	63.6	61.0	**51.8**
ViT-Base	9.5	6.7	8.2	29.0	23.4	33.9	27.1	15.9	26.5	47.2	54.7	44.1	30.5	44.5	47.8	29.9
• Tent	42.2	1.0	43.3	52.4	48.2	55.5	50.5	16.5	16.9	66.4	74.9	64.7	51.6	67.0	64.3	47.7
• SAR	40.8	36.4	41.5	53.7	50.7	57.5	52.8	59.1	50.7	68.1	74.6	65.7	57.9	68.9	65.9	56.3
• EATA	29.7	25.1	34.6	44.7	39.2	48.3	42.4	37.5	45.9	60.0	65.9	61.2	46.4	58.2	59.6	46.6
• COME	51.7	51.4	52.1	57.6	58.2	63.3	41.2	67.1	64.8	72.8	77.7	68.1	67.5	73.0	69.9	62.4
• DeYO	54.0	52.1	55.1	58.8	59.5	64.2	53.5	68.2	66.4	73.7	78.3	68.2	68.9	73.8	70.8	**64.4**
• ZeroSiam (ours)	52.1	52.7	52.8	57.8	58.3	63.0	60.6	67.0	65.7	73.0	77.9	67.6	67.9	72.8	69.9	63.9
ViT-Small	2.0	1.9	1.5	24.3	17.1	30.2	21.8	9.5	19.3	37.9	44.4	29.9	24.9	38.2	41.3	22.9
• Tent	21.3	27.7	22.0	24.5	10.6	32.3	30.9	44.6	52.4	39.6	72.3	44.5	23.4	17.6	55.6	34.6
• SAR	26.9	27.7	23.0	21.7	13.2	26.4	25.3	41.2	45.8	47.7	67.8	39.2	22.5	8.4	33.2	31.3
• EATA	2.4	2.5	1.8	32.8	25.4	36.3	28.2	17.1	28.2	48.7	53.5	46.1	34.1	46.7	47.7	30.1
• COME	0.5	3.5	0.4	48.4	47.3	52.9	15.4	38.2	53.8	64.3	70.7	58.8	55.8	63.6	60.4	42.3
• DeYO	0.4	0.8	0.3	49.0	47.4	53.0	20.1	46.4	54.6	64.7	70.9	59.3	57.3	64.1	61.0	43.3
• ZeroSiam (ours)	26.0	26.9	28.3	48.6	48.2	52.7	50.7	54.4	55.1	65.0	71.0	59.1	58.6	63.4	60.8	**51.3**
ConvNeXt-Tiny	21.3	27.7	22.0	24.5	10.6	32.3	30.9	44.6	52.4	39.6	72.3	44.5	23.4	17.6	55.6	34.6
• Tent	30.3	33.5	32.0	24.7	7.3	33.6	30.0	43.7	47.3	43.7	72.9	49.7	21.4	18.5	56.3	36.3
• SAR	30.3	33.6	29.8	25.9	11.1	33.4	30.8	44.9	47.8	38.3	72.9	46.2	22.6	18.1	56.2	36.1
• EATA	26.6	30.5	26.9	25.2	11.4	33.1	31.0	44.8	50.1	40.0	72.2	47.8	23.5	18.5	55.8	35.8
• COME	41.1	43.2	41.1	37.3	2.1	44.5	17.9	56.4	55.8	60.7	75.2	59.6	8.6	44.7	61.1	43.3
• DeYO	36.8	39.4	37.0	29.5	2.9	38.0	14.6	51.6	46.9	4.5	74.0	55.9	7.8	19.5	57.9	34.4
• ZeroSiam (ours)	41.5	43.6	41.5	37.2	32.5	44.8	42.7	55.6	54.7	62.2	74.3	59.0	42.3	50.5	59.3	**49.4**
Swin-Tiny	26.9	27.7	23.0	21.7	13.2	26.4	25.3	41.2	45.8	47.7	67.8	39.2	22.5	8.4	33.2	31.3
• Tent	23.8	26.2	22.5	19.4	18.4	26.9	27.3	40.4	36.0	22.8	69.2	46.5	16.5	2.4	41.4	29.3
• SAR	25.4	27.1	23.0	21.6	14.8	27.3	27.5	38.9	37.5	44.1	67.7	38.4	18.9	8.3	39.5	30.7
• EATA	32.8	34.5	31.7	23.4	17.5	32.0	28.2	43.5	43.6	45.4	69.7	45.8	27.2	10.5	39.6	35.0
• COME	38.6	41.4	39.0	34.4	14.9	44.6	7.3	52.9	51.0	56.9	72.4	51.7	15.7	33.8	53.3	40.5
• DeYO	35.5	38.5	35.3	22.0	23.9	37.1	15.3	42.4	40.4	47.6	70.2	50.8	23.7	18.0	48.8	36.6
• ZeroSiam (ours)	44.0	45.4	45.2	34.7	39.0	46.3	46.0	53.9	51.3	57.8	72.1	55.0	51.7	50.3	54.6	**49.8**

Table C: Additional results of TTA under **LABEL SHIFTS** with **BATCH SIZE=1** w.r.t. **Accuracy(%)**.

	Noise			Blur				Weather				Digital				
Model+Method	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet50-GN	18.0	19.8	17.9	19.8	11.4	21.4	24.9	40.4	47.3	33.6	69.3	36.3	18.6	28.4	52.3	30.6
• Tent	1.2	1.5	1.3	10.0	2.1	14.3	11.2	8.2	11.7	0.8	70.1	41.6	3.4	49.5	52.3	18.6
• SAR	23.4	26.5	23.8	18.3	15.4	28.4	29.5	44.3	44.5	31.6	72.3	44.5	14.9	46.8	56.1	34.7
• EATA	19.2	21.7	19.4	17.9	13.0	23.5	25.7	39.7	43.6	34.5	69.4	38.4	20.0	34.9	53.2	31.6
• COME	17.8	19.7	17.8	19.7	11.2	21.3	24.9	40.3	47.7	33.6	69.0	35.7	18.6	27.9	51.7	30.5
• DeYO	41.0	43.8	42.2	22.9	23.3	41.4	15.9	54.0	52.3	20.7	73.4	53.6	48.1	60.1	59.8	43.5
• ZeroSiam (ours)	40.3	42.4	40.5	32.6	33.2	41.2	44.3	53.6	53.1	59.5	72.7	52.1	46.8	58.7	58.7	**48.6**
ViT-Base	9.4	6.7	8.3	29.1	23.4	34.0	27.0	15.8	26.3	47.4	54.7	43.9	30.5	44.5	47.6	29.9
• Tent	29.2	7.9	21.1	49.3	46.9	53.7	47.5	18.0	19.2	63.7	71.1	61.4	51.8	63.6	62.2	44.4
• SAR	24.8	18.3	24.1	38.2	31.9	42.1	35.5	31.1	37.5	52.3	61.0	52.3	36.3	52.0	52.4	39.3
• EATA	30.7	26.8	31.3	42.9	39.5	47.8	35.6	38.0	43.7	60.2	65.8	57.7	46.9	59.4	58.0	45.6
• COME	51.1	47.9	52.5	57.2	58.0	63.1	60.5	67.3	64.8	72.8	77.6	67.6	67.4	73.4	69.8	**63.4**
• DeYO	53.2	36.2	54.5	58.2	59.3	64.3	41.8	68.6	66.7	73.8	78.3	67.4	69.3	74.1	71.0	62.4
• ZeroSiam (ours)	49.9	50.2	51.0	56.0	56.4	60.8	58.4	65.5	64.4	71.9	77.4	66.1	65.9	71.5	68.7	62.3
ViT-Small	2.0	2.0	1.5	24.2	17.1	30.3	22.0	9.5	19.2	37.9	44.4	30.1	24.8	38.2	41.0	22.9
• Tent	0.3	0.3	0.2	43.4	37.2	45.7	36.6	4.6	22.3	58.4	66.0	54.7	27.0	56.8	55.3	33.9
• SAR	2.2	2.3	1.7	34.5	26.0	39.2	27.4	19.7	32.1	45.0	53.7	43.4	31.9	44.2	45.9	29.9
• EATA	2.1	2.3	1.7	29.1	21.8	34.6	25.0	14.7	24.8	42.8	49.5	39.4	28.7	42.7	44.2	26.9
• COME	0.5	3.5	0.4	47.8	46.9	52.7	16.9	37.3	53.6	64.3	70.6	58.8	55.8	63.5	60.4	42.2
• DeYO	0.4	0.8	0.3	48.7	47.3	53.2	21.2	46.6	54.2	64.9	70.9	59.4	57.5	64.1	61.0	43.4
• ZeroSiam (ours)	23.7	23.6	25.0	46.0	44.9	49.6	46.8	50.8	52.8	62.7	69.1	56.8	54.8	61.1	58.7	**48.4**
ConvNeXt-Tiny	21.4	27.7	22.1	24.5	11.0	32.5	31.2	44.5	52.5	39.5	72.0	44.8	23.1	17.7	55.8	34.7
• Tent	18.2	24.8	26.0	24.4	4.3	34.0	29.1	41.8	46.3	42.4	73.2	51.8	18.2	10.4	56.9	33.5
• SAR	30.2	33.6	30.2	25.8	11.1	33.3	31.0	44.9	47.6	37.8	73.0	46.2	22.7	18.2	56.3	36.1
• EATA	24.0	28.9	24.5	24.7	11.2	32.5	31.0	44.6	51.4	39.7	72.1	45.9	23.2	18.0	55.8	35.2
• COME	41.4	43.4	41.3	37.3	2.4	44.7	18.4	56.4	55.7	61.1	75.3	59.6	11.0	44.3	61.2	43.6
• DeYO	36.9	39.4	37.0	29.2	3.2	37.7	14.8	51.6	47.6	5.4	74.0	55.8	8.3	23.3	58.0	34.8
• ZeroSiam (ours)	41.3	43.3	41.1	36.3	31.4	43.8	42.2	54.9	53.7	61.5	74.2	58.2	41.0	49.8	59.2	**48.8**
Swin-Tiny	26.6	27.9	22.8	21.7	13.2	26.5	25.4	41.0	45.8	47.7	67.6	39.1	22.6	8.4	33.3	31.3
• Tent	14.6	13.1	11.1	17.4	14.3	21.2	20.9	31.4	24.2	12.1	69.7	48.1	9.1	1.0	42.7	23.4
• SAR	30.5	32.2	29.2	23.0	17.1	29.9	27.9	41.9	39.9	47.5	68.6	40.4	21.5	8.9	40.4	33.3
• EATA	27.7	29.1	25.0	21.7	14.2	27.7	25.9	40.7	43.5	47.4	68.1	39.9	23.4	9.1	34.3	31.9
• COME	38.6	41.0	39.2	34.3	21.6	44.6	5.0	53.1	51.1	57.2	72.2	51.5	21.3	18.3	53.4	40.2
• DeYO	35.7	37.6	35.8	21.0	26.0	36.8	14.7	42.3	42.0	49.6	70.4	50.9	24.4	26.2	48.8	37.5
• ZeroSiam (ours)	42.8	44.4	43.7	33.3	36.1	44.2	43.9	52.1	49.6	55.9	71.6	53.5	49.3	47.7	53.4	**48.1**

Table D: Detailed results of TTA on the **BLIND-SPOT SUBSET**, *i.e.*, Table 2 in the main paper.

	Noise			Blur				Weather				Digital				
Model+Method	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet50-GN	18.0	19.8	17.9	19.8	11.4	21.4	24.9	40.4	47.3	33.6	69.3	36.3	18.6	28.4	52.3	30.6
• Tent	0.2	0.2	0.2	10.5	2.3	5.8	2.3	1.6	2.3	0.2	63.3	16.7	0.5	48.3	46.8	13.4
• SAR	17.0	19.6	16.9	15.8	11.4	22.2	24.0	5.8	42.0	25.4	69.0	38.9	1.3	31.9	54.2	26.4
• EATA	18.6	21.2	18.6	15.6	13.4	22.7	24.4	37.1	38.8	33.2	67.8	41.6	19.1	47.1	53.0	31.5
• COME	18.0	19.8	17.9	19.8	11.3	21.4	24.9	40.4	47.3	33.6	69.2	36.2	18.6	28.3	52.2	30.6
• DeYO	0.4	0.6	0.4	7.9	0.2	28.1	1.6	5.5	7.1	0.1	71.2	47.8	1.3	59.7	51.4	18.9
• ZeroSiam (ours)	38.0	42.7	39.0	27.9	34.0	41.1	43.3	52.1	44.3	59.1	60.2	51.2	53.8	60.3	45.4	**46.2**
ViT-Base	9.5	6.7	8.2	29.0	23.4	33.9	27.1	15.9	26.5	47.2	54.7	44.1	30.5	44.5	47.8	29.9
• Tent	0.2	0.2	0.1	56.5	54.0	60.4	0.5	1.4	1.0	0.2	77.5	67.4	0.3	71.6	68.7	30.7
• SAR	42.2	7.1	43.9	55.3	51.3	59.4	54.0	19.7	47.6	71.5	77.4	67.1	19.4	71.3	67.8	50.3
• EATA	26.4	20.3	32.2	45.7	37.5	48.6	42.3	37.2	47.5	62.2	65.7	63.1	46.8	60.1	59.8	46.4
• COME	0.1	0.1	0.1	56.9	58.4	63.5	0.5	68.3	64.7	73.4	77.1	67.1	69.1	73.4	69.5	49.5
• DeYO	0.1	0.3	0.2	59.4	60.4	65.1	3.8	69.7	67.6	74.5	78.5	68.7	70.8	75.0	71.2	51.0
• ZeroSiam (ours)	52.6	53.2	53.4	57.7	58.6	63.4	61.7	67.8	66.5	73.8	78.2	67.8	69.2	73.7	70.2	**64.5**
ViT-Small	2.0	1.9	1.5	24.3	17.1	30.2	21.8	9.5	19.3	37.9	44.4	29.9	24.9	38.2	41.3	22.9
• Tent	0.1	0.2	0.1	44.6	36.6	45.6	33.2	1.3	16.5	59.5	68.0	56.5	0.5	58.2	56.2	31.8
• SAR	2.1	2.2	1.6	31.1	23.7	39.2	27.5	9.8	40.0	51.8	66.4	46.7	23.4	53.2	49.8	31.2
• EATA	2.2	2.2	1.6	28.4	20.3	33.2	24.2	11.8	23.0	44.8	51.4	47.4	28.3	45.4	45.6	27.3
• COME	0.1	2.6	0.1	47.3	48.0	52.4	0.2	31.2	55.6	65.0	71.7	59.5	0.4	64.4	60.6	37.3
• DeYO	0.1	0.3	0.1	49.4	48.9	54.0	0.5	45.9	56.3	66.2	72.3	60.6	58.6	65.5	61.9	42.7
• ZeroSiam (ours)	25.1	25.4	27.6	48.2	47.9	52.5	50.6	54.3	55.5	65.6	71.8	59.2	59.4	64.6	61.2	**51.3**
ConvNeXt-Tiny	21.3	27.7	22.0	24.5	10.6	32.3	30.9	44.6	52.4	39.6	72.3	44.5	23.4	17.6	55.6	34.6
• Tent	15.5	12.9	25.2	21.3	2.1	30.0	25.3	30.3	40.4	1.8	72.6	50.1	14.3	3.2	56.4	26.8
• SAR	24.3	30.2	24.8	24.7	10.7	32.5	30.6	42.9	50.1	37.1	72.7	45.1	21.8	17.7	55.7	34.7
• EATA	23.7	29.1	24.5	24.6	11.1	32.5	30.7	44.3	50.4	39.9	72.0	46.2	23.2	17.9	55.7	35.0
• COME	41.1	42.7	40.4	29.4	0.4	33.3	2.4	25.1	24.9	0.2	74.2	57.7	0.6	27.6	58.3	30.6
• DeYO	34.9	35.5	35.3	22.0	0.9	23.0	3.0	38.6	29.1	0.3	72.1	52.3	1.8	22.1	55.6	28.4
• ZeroSiam (ours)	42.4	44.9	42.5	37.2	33.9	45.5	44.5	56.3	53.3	64.4	73.7	60.5	44.7	56.3	58.5	**50.6**
Swin-Tiny	26.9	27.7	23.0	21.7	13.2	26.4	25.3	41.2	45.8	47.7	67.8	39.2	22.5	8.4	33.2	31.3
• Tent	0.2	0.4	0.1	14.0	5.9	7.2	5.0	9.6	8.9	2.6	67.7	16.1	1.4	0.3	34.4	11.6
• SAR	27.3	29.0	24.8	21.6	14.9	26.7	25.7	36.6	45.3	47.5	67.8	39.4	10.5	8.6	36.5	30.8
• EATA	27.9	29.0	25.3	20.7	14.6	28.4	25.7	40.1	40.7	47.4	68.1	40.8	23.6	9.2	33.8	31.7
• COME	38.1	41.3	42.0	26.2	0.5	10.8	0.5	10.3	8.0	10.2	72.8	52.0	0.1	0.5	55.1	24.6
• DeYO	0.2	0.1	0.2	13.8	6.4	11.4	3.7	11.6	16.9	4.3	70.2	44.4	3.7	0.4	47.1	15.6
• ZeroSiam (ours)	45.1	47.5	47.0	35.2	40.3	46.1	46.9	56.1	51.9	58.5	68.6	54.8	54.7	52.9	56.3	**50.8**

Table E: Detailed results of TTA under the **MILD SCENARIO**.

	Noise			Blur				Weather				Digital				
Model+Method	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet50-GN	18.0	19.8	17.9	19.8	11.4	21.4	24.9	40.4	47.3	33.6	69.3	36.3	18.6	28.4	52.3	30.6
• Tent	5.1	6.1	5.4	15.0	10.8	22.1	22.9	25.8	33.2	3.1	70.4	42.7	11.0	48.1	54.2	25.1
• SAR	28.5	31.4	29.7	18.7	19.5	30.2	30.1	43.3	43.7	7.0	70.8	44.0	19.0	48.8	55.2	34.7
• EATA	37.3	39.1	39.4	28.1	27.1	36.8	39.1	50.8	49.0	55.8	72.0	50.1	41.8	55.7	58.1	45.4
• COME	17.9	19.7	17.7	19.7	11.3	21.4	24.9	40.4	47.5	33.6	69.1	35.9	18.6	27.9	52.1	30.5
• DeYO	39.6	42.1	40.6	22.2	23.3	38.3	37.7	50.4	49.4	3.0	73.0	50.3	41.9	55.5	57.7	41.7
• ZeroSiam (ours)	40.5	42.4	41.7	31.4	31.8	40.8	44.4	53.5	53.3	59.1	72.9	51.5	47.4	58.9	58.8	**48.6**
ViT-Base	9.4	6.7	8.3	29.1	23.4	34.0	27.0	15.8	26.3	47.4	54.7	43.9	30.5	44.5	47.6	29.9
• Tent	42.4	1.4	43.3	52.2	47.6	55.4	49.9	19.7	18.1	66.1	74.9	64.7	52.5	66.8	64.1	47.9
• SAR	44.3	14.0	45.7	52.9	49.8	56.0	51.1	58.3	50.6	66.5	74.6	64.3	55.5	66.5	64.0	54.3
• EATA	49.7	49.4	51.3	55.9	55.5	60.5	57.8	63.4	63.0	70.4	76.0	67.0	64.7	69.3	67.9	61.5
• COME	50.9	51.6	52.1	56.9	57.4	62.2	59.2	66.1	64.3	72.5	77.4	67.9	66.3	72.4	69.2	**63.1**
• DeYO	52.9	53.1	53.8	58.2	58.6	63.0	40.2	67.4	65.7	73.2	78.0	68.0	67.7	73.1	69.8	62.8
• ZeroSiam (ours)	50.5	51.0	51.5	57.1	56.8	61.2	58.6	64.9	64.5	71.9	77.2	67.6	65.6	71.5	68.5	62.6
ViT-Small	2.0	2.0	1.5	24.2	17.1	30.3	22.0	9.5	19.2	37.9	44.4	30.1	24.8	38.2	41.0	22.9
• Tent	0.5	0.6	0.4	39.5	33.0	42.5	34.0	10.3	37.1	54.5	63.6	51.7	31.5	53.1	52.3	33.6
• SAR	1.4	2.1	1.1	40.8	34.9	43.2	35.5	16.3	41.8	54.7	63.7	51.8	38.5	53.4	52.5	35.4
• EATA	22.4	20.6	23.6	45.0	43.0	47.2	43.5	48.5	49.9	60.7	66.4	56.6	52.7	58.3	57.2	46.4
• COME	0.2	0.4	0.2	47.7	46.3	51.3	19.6	37.9	52.7	63.6	69.8	58.3	54.1	61.8	59.2	41.6
• DeYO	0.2	0.3	0.2	47.9	45.8	51.4	33.6	44.5	53.0	63.5	70.1	58.5	55.3	62.4	59.8	43.1
• ZeroSiam (ours)	23.8	23.4	25.2	47.6	46.0	50.6	48.1	51.9	53.3	63.2	69.9	58.1	55.7	61.8	59.4	**49.2**
ConvNext-Tiny	21.4	27.7	22.1	24.5	11.0	32.5	31.2	44.5	52.5	39.5	72.0	44.8	23.1	17.7	55.8	34.7
• Tent	30.8	33.6	32.0	24.7	7.5	33.6	30.0	43.7	47.3	43.6	72.9	49.5	21.5	19.3	56.3	36.4
• SAR	33.4	35.6	33.7	27.1	8.1	34.1	30.7	44.6	47.3	17.0	72.8	48.8	22.0	24.4	56.1	35.7
• EATA	35.3	37.4	35.8	31.0	20.7	37.1	33.7	48.5	49.1	51.3	73.2	53.0	29.5	31.3	56.6	41.6
• COME	40.4	42.6	40.4	36.2	2.3	43.2	23.4	54.6	54.2	58.1	75.1	58.6	10.6	44.7	59.9	43.0
• DeYO	36.3	39.0	36.4	29.1	2.7	37.1	19.6	50.1	48.3	4.8	73.8	54.9	8.6	23.7	57.4	34.8
• ZeroSiam (ours)	39.6	41.7	39.3	33.8	27.3	42.0	38.7	53.2	51.8	59.3	73.8	57.0	36.5	44.2	58.2	**46.4**
Swin-Tiny	26.6	27.9	22.8	21.7	13.2	26.5	25.4	41.0	45.8	47.7	67.6	39.1	22.6	8.4	33.3	31.3
• Tent	23.7	26.2	22.7	19.4	18.6	27.1	27.4	40.5	36.3	24.4	69.2	46.3	16.9	2.8	41.5	29.5
• SAR	31.8	32.8	32.5	22.3	20.2	29.6	28.4	42.6	39.8	37.6	69.2	46.8	19.9	6.0	43.2	33.5
• EATA	40.9	42.9	41.6	33.5	33.6	42.6	40.1	50.1	49.1	54.9	71.8	52.6	44.8	40.0	51.1	46.0
• COME	41.2	43.2	42.1	35.2	15.0	44.7	7.0	52.9	51.9	57.8	72.1	51.1	23.8	25.2	54.3	41.2
• DeYO	35.2	37.0	36.0	22.2	25.0	35.8	15.4	42.0	43.8	48.0	70.1	50.3	31.6	34.5	48.0	38.3
• ZeroSiam (ours)	42.4	44.0	43.0	31.7	34.9	43.6	42.6	51.3	48.9	54.7	71.5	53.4	47.5	45.8	52.3	**47.2**

E ADDITIONAL DISCUSSIONS

E.1 ZEROSIAM'S EFFICACY BEYOND COLLAPSE PREVENTION

ZeroSiam regularize test-time entropy minimization from favoring non-generalizable shortcuts (c.f. Section 2.2), thereby improving its performance even in cases when no collapse occurs. As shown in Figure B, Tent drastically reduces entropy and eventually closes it to zero by inflating the logit norms and aligning all logits toward a dominant mode. Despite the reduction of entropy, it does not involve meaningful learning, and Tent thus fails to improve the accuracy on the first 250 batches of samples, where such biased signals can further degrade performance during a prolonged adaptation, as shown in Figure B. In contrast, ZeroSiam maintains a stable logit norm and center dominance ratio during entropy minimization, which successfully enhances TTA efficacy beyond collapse prevention, as shown in Figure B (a). Moreover, from Figure B (d), ZeroSiam does not greedily minimize the entropy loss. Instead, ZeroSiam converges to a low but non-zero entropy loss, while still enabling consistent accuracy improvement, *i.e.*, from 250 to 750 batches as in Figure B (a). Such convergence toward non-zero loss is also aligned with Theory 1, which shows that ZeroSiam inherently defines a lower bound of h_{min} for entropy optimization, preventing the model from degenerating into collapsed constant one-hot outputs that trivially minimize (*i.e.*, zero out) the entropy loss.

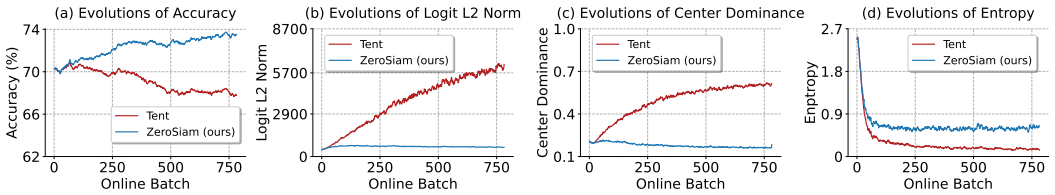

Figure B: Empirical evidence of ZeroSiam for boosting stability and efficacy. (a-d) record the ODD accuracy, logits L_2 norm, center dominance, and entropy in model predictions under a mild test scenario (Wang et al., 2021). Experiments are run on ImageNet-C (Bright, level 5) with ViT-Base.

E.2 EVOLUTIONS OF DIVERGENCE LOSS IN ZEROSIAM

We provide additional results of how the divergence loss evolves during adaptation to supplement Figure 2. As shown in Figure C, the divergence loss increases more rapidly and substantially under a more imbalanced stream. Such a phenomenon is also aligned with changes of the predictor as shown in Figure 2 (a), where a larger divergence from an identity mapping results in a larger similarity loss. Overall, these results suggest *an adaptive regularization strength* according to the degree of collapse risk in the scenario, indicating the efficacy of our asymmetric entropy optimization design.

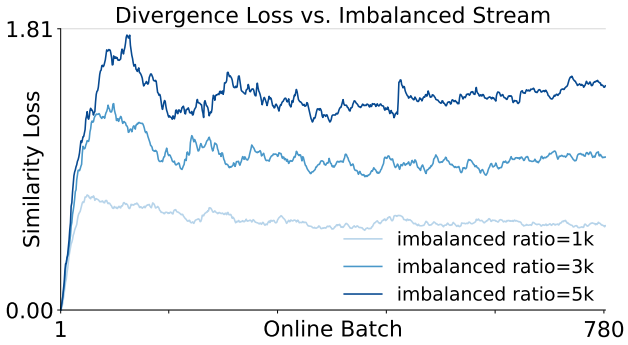

Figure C: The evolution of divergence loss during TTA . Results are reported on ImageNet-C (Snow, level 5) with ResNet50-GN under online streams with varying imbalanced ratios (Niu et al., 2023).

E.3 ZEROSIAM FOR SAVING A COLLAPSED MODEL

Interestingly, we further demonstrate that ZeroSiam can sometimes save a model that has already collapsed. To illustrate this, we first run Tent on an imbalanced data stream until collapse occurs, and then apply ZeroSiam. As shown in Figure D, ZeroSiam helps restore strong accuracy after 300 batches of adaptation. This recovery is driven by the asymmetric alignment in ZeroSiam, which makes collapsed solutions no longer a stable minimum, encouraging the model to escape the collapsed mode and re-cluster samples. However, this does not fully explain how class-wise clusters that are consistent with pre-training re-emerge. We hypothesize this is because only the affine parameters in the normalization layers are updated during testing, which introduces a small adjustment to the model that makes performance recovery possible. Notably, we observe that the predictor in ZeroSiam has to be randomly initialized (*e.g.*, $\theta_h = \mathbf{I} + 0.1\,\mathbf{W}$, $\mathbf{W} \sim \mathcal{N}(0, \mathbf{I})$) so as to derive from the identity and also remain learnable during TTA to enable the collapse recovering effects. Even so, recovery succeeds in only 4 out of 7 domains, suggesting that while promising, collapse recovery with ZeroSiam remains an open question, and we leave it for future work.

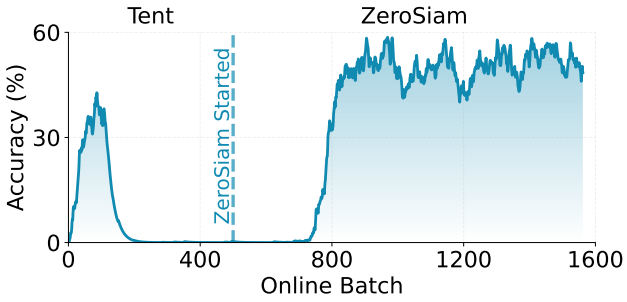

Figure D: Efficacy of ZeroSiam for saving a collapsed model. Results are reported on ImageNet-C (Shot, level 5) with ViT-Base under **ONLINE IMBALANCED LABEL SHIFTS** (imbalance ratio = ∞).

E.4 COMPARISONS BETWEEN THE ENTROPY AND DIVERGENCE TERM DURING TTA

As shown in Figure E, the alignment loss $D(p^o||\text{sg}[p^r])$ does not grow infinitely. Instead, the alignment loss quickly increases then converges, while the entropy term $H(p^o)$ decreases and stabilizes. Throughout adaptation, the alignment term exhibits a smaller loss value than the entropy term, which provides an appropriate regularization strength that prevents the risk of collapse, without hindering the effectiveness of the entropy-based adaptation.

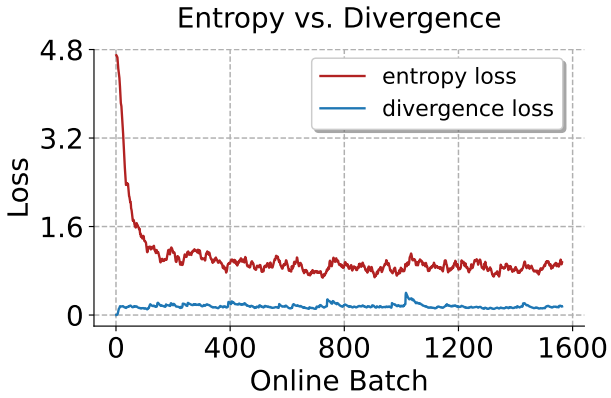

Figure E: Comparisons between entropy and divergence during TTA on ImageNet-C (Gaussian, level 5) under **ONLINE IMBALANCED LABEL SHIFTS** (imbalance ratio = ∞) with ViT-Base.

E.5 More Sensitivity Analysis of Learning Rates in ZeroSiam

To demonstrate ZeroSiam's robustness, we further supplement the sensitivity analysis of learning rates on Swin-Tiny, a challenging architecture on which Tent frequently suffers from collapse.

From Figure F, ZeroSiam consistently outperforms Tent across a wide range of encoder learning rates. Moreover, on smaller and more collapse-prone models, ZeroSiam can benefit from a larger predictor learning rate η_h, *e.g.*, yielding an accuracy of 53.3% with $\eta_f = 10 \times 5e-5$ and $\eta_h = 40 \times \eta_f$; while still outperforming previous methods across broad learning rate configurations, *e.g.*, achieving an accuracy above 45% when $\eta_f \in [5, 20] \times 5e-5$ and $\eta_h \in [1, 40] \times \eta_f$, showing a substantial gain compared to the accuracy of 38.2% in COME and 37.7% in DeYO. These results collectively demonstrate the efficacy of ZeroSiam across learning rate choices. In practice, we did not extensively tune the learning rates, where we simply use $\eta_f = 10 \times 5e-5$ and $\eta_h = 5 \times \eta_f$ on ZeroSiam, which yields an accuracy of 50.4% for comparisons with other methods.

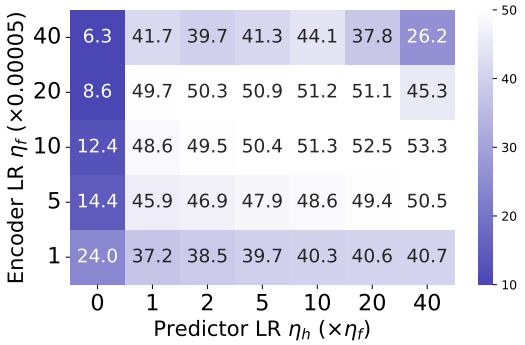

Figure F: Sensitivity to learning rates on Swin-Tiny. Results are reported on ImageNet-C (Gaussian, level 5) under **ONLINE IMBALANCED LABEL SHIFTS** (imbalance ratio = ∞).

E.6 Discussions on a *Data-Free* Approach for Selecting η_h in ZeroSiam

While the predictor lr η_h in ZeroSiam is simply set the same as η_f on larger models and around $10\times$ larger than η_f on smaller and more collapse-prone models, we further briefly introduce **a data-free method** to help select η_h. The core idea is to leverage random Gaussian noise as the proxy to track TTA dynamics, where ideally, noisy learning signals should be totally absorbed and suppressed in ZeroSiam. To this end, we track the changes in prediction, *e.g.*, prediction entropy Δ_e, before and after adaptation on Gaussian noise, and select the best-performing η_h that leads to a minimal change, *e.g.*, $\min_{\eta_h} |\Delta_e|$, which implies an appropriate regularization strength for bias filtration. As shown in Table V, on ResNet50-GN, we can find $\eta_h = 10$ by $\min_{\eta_h} |\Delta_e|$ using only generated Gaussian before TTA deployment, which also delivers the highest TTA accuracy of 51.4% under label shift.

Table F: Illustration of data-free selection for η_h on ResNet50-GN based on entropy difference $|\Delta_e|$ before and after adaptation on 6,400 random Gaussian inputs. TTA accuracy is measured under the label shift setting. Encoder learning rate η_f is set to 0.00125, and η_h is set to $k\times$ 0.00125.

Metric	$\eta_h = 0$	$\eta_h = 2$	$\eta_h = 5$	$\eta_h = 10$	$\eta_h = 20$	$\eta_h = 40$
prediction $\lvert\Delta_e\rvert$	0.750	0.085	0.053	0.001	0.061	0.083
TTA Acc. (%)	14.3	49.6	51.0	51.4	50.3	45.3

We acknowledge that some architectures, such as ConvNeXt, can consistently output the highest prediction entropy on Gaussian noise regardless of adaptation and may require tracking other prediction metrics for data-free η_h selection. We leave this for future work, and believe that our current focus on collapse prevention and asymmetric entropy-based architecture already makes meaningful contributions, as this significantly reduces the risk of TTA across adaptation scenarios while maintaining efficiency for practical and reliable deployment.

E.7 EFFICACY OF ZEROSIAM ACROSS DIVERSE ADAPTATION EPOCHS

In our main experiments, all methods are evaluated with only 1 epoch over the test set, following the evaluation settings established by prior works. Here, we further evaluate Tent and ZeroSiam using more adaptation epochs under Tent's default mild setting. As shown in Table G, Tent's performance gradually degrades with more epochs (47.9% $\rightarrow$ 41.3%), reflecting its known overfitting tendency. In contrast, ZeroSiam remains stable and even benefits moderately from additional epochs (62.6% $\rightarrow$ 65.7%). This confirms that ZeroSiam improves generalization while mitigating overfitting in Tent, which maintains a stable adaptation across different epoch budgets.

Table G: Accuracy of Tent and ZeroSiam under the k-th epoch of TTA over the entire test set. Results are reported with ViT-Base under Tent's default mild setting.

Method	No adapt	Tent (k=1)	Tent (k=3)	Tent (k=5)	ZeroSiam (k=1)	ZeroSiam (k=3)	ZeroSiam (k=5)
Acc. (%)	29.9	47.9	45.5	41.3	62.6	65.7	65.7

E.8 NECESSITY OF STOP-GRADIENT FOR ASYMMETRY

As verified in Theory 1, the online branch in ZeroSiam converges more rapidly towards the collapse solutions during TTA. The stop-gradient on the target branch prevents the target branch from also drifting toward the collapse modes, while providing a stable reference for the online branch. As shown in Table H, removing stop-gradient leads to severe collapse, *e.g.*, reducing the accuracy from 51.6%$\rightarrow$20.5% on ResNet50-GN. Similar results are also observed in SimSiam (Chen & He, 2021) and BYOL (Grill et al., 2020), underscoring stop-gradient as a key component in asymmetry.

Table H: Importance of stop-gradient operation. Results are reported on ImageNet-C (severity level 5) under **ONLINE IMBALANCED LABEL SHIFTS** (imbalance ratio = ∞) regarding **Accuracy (%)**.

	Noise			Blur				Weather				Digital				
ResNet50-GN	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ZeroSiam (ours)	42.9	45.1	43.1	35.1	36.1	44.5	49.2	58.1	55.1	62.9	72.3	54.8	52.5	61.7	60.1	51.6
- w/o stop-grad	2.1	2.7	2.0	15.3	2.0	15.2	11.1	11.9	17.7	1.2	71.9	43.5	3.8	50.9	55.5	20.5
ViT-Base	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ZeroSiam (ours)	52.3	52.6	53.4	57.7	58.7	62.7	61.1	67.6	66.0	73.3	78.0	67.0	67.9	73.5	70.1	64.1
- w/o stop-grad	2.8	0.8	12.8	54.1	51.1	57.6	33.7	6.3	8.0	68.7	76.1	65.9	4.4	69.2	66.3	38.5

E.9 STATISTICAL COMPARISON

We re-run Table 3 in the main paper with 5 different random seeds. As shown in Table I, ZeroSiam achieves both higher adaptation accuracy and lower standard deviation compared to prior methods. Specifically, ZeroSiam increases the average accuracy from 42.6% (DeYO) to 52.9%, while reducing the standard deviation by over *10*-fold. This reduced standard deviation further underscores our reliability for robust adaptation in real-world applications.

Table I: Statistical comparison. Experiments follow the settings of Table 3 in the main paper. Results are reported with 5 random seeds.

Method	ResNet50-GN	ViT-Base	ViT-Small	ConvNeXt-Tiny	Swin-Tiny	Average
DeYO	$39.7_{\pm 2.05}$	$61.8_{\pm 2.19}$	$40.8_{\pm 0.36}$	$34.3_{\pm 0.40}$	$36.5_{\pm 0.71}$	$42.6_{\pm 0.82}$
ZeroSiam	$51.2_{\pm 0.06}$	$63.5_{\pm 0.10}$	$50.0_{\pm 0.06}$	$49.8_{\pm 0.11}$	$50.2_{\pm 0.16}$	$52.9_{\pm 0.06}$

E.10 MORE DISCUSSIONS ON THE IMPACTS OF UNCONTROLLED LOGIT NORM INFLATION

During test-time entropy minimization, uncontrolled inflation of logit norm can induce the following issues that hinders robust and reliable model adaptation: 1) *Overconfidence*: regarding individual predictions, increasing the logit magnitude $||u||_2$ (where u is the predicted logit) artificially

inflates predictive confidence without improving accuracy, since the predicted class $\arg\max_k u_k$ depends solely on the direction $u' = \frac{u}{||u||_2}$, not its magnitude. This leads to overly confident but inaccurate predictions that undermine trust in AI systems. 2) *Degraded generalization from gradient interference*: while inflating the logit norm does not benefit a sample's own generalization, it adversely affects others through gradient interference during batch updates. Empirically, test-time adaptation with only the objective of $-||u||_2$ to inflate the logit norm would lead to a catastrophic collapse of zero accuracy for all models. Theoretically, define the decision margins as $m(x) = \max_k u_k - \max_{j \neq k} u_j$ and the sharpness of relative boundaries $\frac{||\nabla_x m(x)||}{m(x)}$, given that both $m(x)$ and $||\nabla_x m(x)||$ scale proportionally with $||u||_2$, inflating the logit norm can not improve the sample's own relative boundary. However, since per-example gradients are generally non-orthogonal, a sample's update can *add irrelevant noise to the boundaries of other samples while having no benefit on its current prediction*. Such interference accumulates during online TTA, adversely affecting generalization and ultimately causing collapse. ZeroSiam mitigates this by aligning with a stable target branch, preventing uncontrolled norm growth, and improving TTA efficacy.

E.11 More Discussions with Multi-Branch Adaptation Methods

ZeroSiam advances the entropy-based, self-training TTA by adding both the mechanisms of bias learning signals filtration (c.f. Section 2.2) and asymmetric optimization (c.f. Section 2.1 through a lightweight predictor with theoretical insights (Theorem 1). We further detail our distinctions with multi-branch-based TTA methods, such as SPA (Niu et al., 2025), REM (Han et al., 2025), and TTE (Kim et al., 2025) from three aspects.

(1) *Distinct inspirations from self-supervised learning (objective vs. architecture):* SPA and REM are augmentation-based methods that exploit consistency learning—an objective similar to self-supervised approaches—across different views to promote adaptation to the target domain. To this end, their efforts focus on devising appropriate augmentations or deteriorations on images for information masking and creating a weak-to-strong consistency learning. In contrast, ZeroSiam is entirely augmentation-free and focuses on the asymmetric architecture design—orthogonal to efforts in SPA and REM—for the single-branch, entropy-based adaptation, not being limited to images.

(2) Advantages of model collapse prevention (model-, test data stream- & modality-agnostic): REM prevents collapse by depending on the assumption that masked images should exhibit higher prediction entropy, which develops architecture-specific masking strategies for the vision transformer equipped with a classification token and explicitly fixes the rank of prediction entropy of masked inputs. However, the assumption and the masking strategy may fail to generalize to broader architectures and modalities, while the objective can still be minimized with constant one-hot outputs. TTE prevents collapse by subtracting a moving average center in the predicted logit for self-alignment, which explicitly reduces center dominance in a straightforward manner. However, it assumes that the prediction center truly reflects the trends of overfitting, which, however, violates under a long-tail data stream where the center actually reflects the dominant class. In this case, when subtracting the center, TTE can significantly risk flipping correct predictions and producing misguided alignment. In contrast, ZeroSiam develops an assumption-free and theoretically grounded asymmetric architecture that self-induces collapse-resistant alignment to disfavor shortcuts in entropy minimization without relying on heuristic thresholds, making it generally applicable across diverse models, data streams, and modalities and requiring fewer hyperparameters for deployment.

(3) Unique strengths of adaptive bias filtration (improving generalization): Beyond the widely applicable collapse prevention mechanism in ZeroSiam compared to prior methods, we also empirically and theoretically demonstrate that ZeroSiam further improves generalization with a bias learning signals filtration mechanism during TTA. Specifically, we show that the predictor in the online branch purposely absorbs and converts biased shortcut signals (*e.g.*, norm inflation) into explicit discrepancies, where biased signals are then penalized by the alignment loss. As shown in Theorem 1 and Figure 2 (b-c), any shortcut representable by the predictor is naturally suppressed, while useful signals flow into the backbone encoder update. This bias filtration effect even helps improve the generalization of a large language model during test-time reasoning incentivization, demonstrating a promising research direction to enhance future TTA methods.

Finally, we provide further empirical comparisons on both TTA performance and efficiency. As shown in Table J, existing methods improve performance at a substantial cost of computation and

memory efficiency, due to using additional augmentations and branches, *e.g.*, SPA increases the memory of Tent from 7,760MB to 28,147MB (which can not be fitted into an RTX 3090), and the latency from 113s to 275s on ViT-Base. In contrast, ZeroSiam is the only method that maintains almost the same memory and latency as Tent, while achieving the highest stability, *e.g.*, increasing the accuracy of SPA from 60.3% to 64.5%, while reducing the memory from 28,147MB to 7,806MB on ViT-Base, which reveals a novel and practical stability–efficiency Pareto frontier that has not been achieved in prior TTA methods.

Table J: comparisons on performance and efficiency with multi-branch TTA methods. Accuracy is reported under the blind-spot adaptation scenario. Efficiency is measured by processing 50,000 images under a batch size of 64 via an A100 GPU.

	ResNet50-GN			ViT-Base		
Method	Acc. (%)	Mem. (MB)	Time (s)	Acc. (%)	Mem. (MB)	Time (s)
Tent	13.4	5,519	89	30.7	7,760	113
TTE	40.8	6,303	136	55.2	8,574	164
REM	-	-	-	56.9	27,438	346
SPA	45.0	21,327	220	60.3	28,147	275
ZeroSiam (ours)	46.2	5,584	89	64.5	7,806	113

F More Discussions with Reinforcement Learning

Entropy collapse is also a common challenge within reinforcement learning, where policies can degenerate (*e.g.*, become deterministic and lose diversity) due to biased or noisy advantage estimation, insufficient exploration, or overly aggressive updates (Mnih et al., 2016; Haarnoja et al., 2018; Ahmed et al., 2019). To address this, trust-region methods such as TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017) constrain each update step in policy space through KL-divergence penalties or clipping, ensuring gradual and stable improvement. Similar principles appear in value-based methods (Mnih et al., 2015; Schwarzer et al., 2023; Nauman et al., 2024), where target networks provide a stable anchor to stabilize bootstrapped updates against rapidly changing estimates. These approaches share with ZeroSiam the key idea of structurally constraining updates to counter collapse. Specifically, ZeroSiam's online branch with gradient updates resembles a fast-updating policy (*i.e.*, with extra updates in the predictor), while the stop-gradient target branch serves as an anchor, effectively forming a lightweight trust region in the entropy minimization landscape. This connection situates ZeroSiam as a general collapse-prevention mechanism, linking stabilization strategies across TTA, self-supervised learning, and reinforcement learning.

G Large Language Model Usage Statement

In accordance with the ICLR 2026 policy on the responsible use of LLMs, we confirm that our study did not use any LLM to generate scientific content or conduct substantive experiments. The only use of an LLM (ChatGPT-5) was to polish the English writing and improve presentation quality; all core methodology, experiments, and analyses were authored and verified by the human authors.