TEST-TIME CO-ADAPTATION AS DATA ASSIMILATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time adaptation under corruption mainly follows two directions: directly adapting models to corrupted inputs, or restoring inputs before prediction. These two routes are complementary, yet simply combining them does not necessarily outperform model adaptation alone, since corrupted inputs may lose discriminative evidence while restoration can introduce biased or misleading cues. Inspired by data assimilation, we treat the corrupted input as an observation and its diffusion-restored counterpart as a prior estimate, and propose *test-time co-adaptation* (**CoDA**) to address this issue. CoDA regulates cross-view interaction through cross-view gating for reliable updates, selective Jensen–Shannon coupling for consistent samples, and entropy-aware logit fusion for prediction, while leaving the underlying TTA objective unchanged. On ImageNet-C at severity 5, across five TTA methods and five backbone architectures, CoDA improves independent data–model adaptation by 2.8 percentage points on average and up to 4.4 points, reaching **54.7%** accuracy with ZeroSiam. The gains persist under label shift with batch size one and continual adaptation, and extend to natural rendition shifts on ImageNet-R and geometry-aware corruptions on ImageNet-3DCC, showing that CoDA is effective beyond standard 2D corruptions and generalizes to broader natural and structured distribution shifts.

1 INTRODUCTION

Test-time adaptation (TTA) adapts a source-trained classifier to unlabeled target inputs at deployment. Under corruption, the same inputs used for adaptation may already carry degraded class evidence. Existing work mainly follows two lines: *model adaptation*, which updates the classifier on corrupted target inputs through entropy minimization, confidence filtering, consistency regularization, and related objectives (Wang et al., 2020; Niu et al., 2022; 2023; Lee et al., 2024; Zhang et al., 2025; Chen et al., 2026), and *data adaptation*, which restores each input with diffusion models (Ho et al., 2020; Song et al., 2020; 2021; Meng et al., 2022) before prediction (Gao et al., 2023; Tsai et al., 2024; Guo et al., 2025). These approaches have largely developed as separate adaptation routes. Although diffusion models have also been used to guide model updates (Prabhudesai et al., 2023; Li et al., 2024; Yang et al., 2026), a central question remains: how should a classifier learn from both a corrupted observation and its diffusion-restored counterpart?

Model adaptation and data adaptation provide complementary information but exhibit distinct limitations. Adapting only to corrupted inputs fails to exploit the visual structure recovered by restoration and may reinforce erroneous predictions when discriminative evidence is weak. Conversely, diffusion-based data adaptation keeps the classifier fixed despite the restoration-induced distribution shift, while the restored image is only a prior-informed estimate (Chung et al., 2023; Zhu et al., 2023; Saharia et al., 2022; Lugmayr et al., 2022; Chihaoui et al., 2024) that may suppress class-discriminative details or introduce misleading cues. Neither view is uniformly reliable: the corrupted observation retains sample-specific evidence that restoration may alter, whereas the restored view provides structural information from the learned image prior. This complementarity motivates jointly adapting to both views.

We first consider a simple *dual-path data–model adaptation* scheme, where two classifier branches are adapted to the corrupted observation and the restored view, and their predictions are fused at inference time. Although this scheme substantially improves over data adaptation alone, it does not consistently outperform model adaptation on corrupted inputs (Figure 2). Because the two branches adapt to different test-time distributions, independent updates can drive their predictions apart, while

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

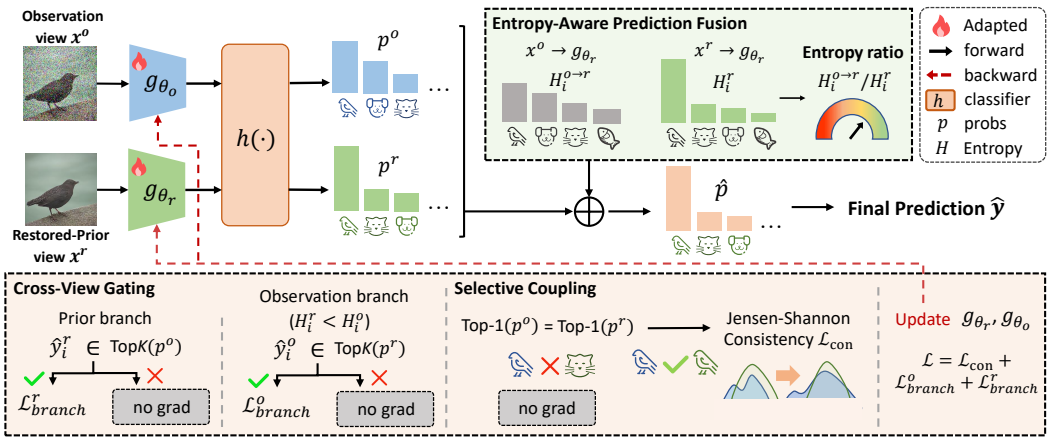


Figure 1: **Overview of CoDA.** The corrupted image is treated as the observation and its restoration as a prior-informed estimate. Cross-View Gating controls branch updates through cross-view top-k support (Sec. 4.1); Selective Coupling applies Jensen–Shannon consistency under top-1 agreement (Sec. 4.2); and Entropy-Aware Prediction Fusion combines branch logits (Sec. 4.3).

naive fusion may further weaken the stronger branch. Thus, the key challenge is not merely to exploit both views, but to effectively control their interaction during adaptation and prediction.

Inspired by *data assimilation*, which reconciles observations with model-based prior information (Kalnay, 2003; Evensen, 2009; Houtekamer & Mitchell, 1998), we treat the corrupted image as an observation and its diffusion restoration as a prior-informed estimate. Their predictive mismatch motivates a *semantic innovation* view, realized in CoDA through cross-view compatibility.

We instantiate this view with *test-time co-adaptation* (**CoDA**), as shown in Figure 1. CoDA adapts two branches to the corrupted and restored views. Cross-view support filters prior-branch updates and conditionally constrains the observation branch. Samples that further agree at top-1 are coupled with a symmetric Jensen-Shannon term to limit cross-branch drift. At inference, the two branch logits are combined according to their entropy-based reliability. The branch-specific TTA objective is preserved; CoDA changes the adaptation loop instead.

We evaluate CoDA across five TTA methods and five backbone architectures on ImageNet-C (Hendrycks & Dietterich, 2019) at severity 5. Compared with independent DDA+TTA, CoDA improves accuracy by 2.8 percentage points on average and up to 4.4 points, reaching 54.7% with ZeroSiam (Chen et al., 2026). The gains extend to label shift under batch size one, and continual adaptation. Further evaluations on ImageNet-R and ImageNet-3DCC demonstrate improvements under natural rendition and geometry-aware corruption shifts. Our contributions are as follows.

- We formulate dual-path TTA over corrupted observations and diffusion-restored prior-informed estimates, highlighting the need for controlled adaptation between the two views.
- Inspired by data assimilation, we propose CoDA, a controlled co-adaptation protocol that treats the corrupted input as an observation and its restoration as a prior, regulating their interaction through cross-view update gating, selective Jensen–Shannon coupling, and entropy-aware prediction fusion.
- Extensive experiments across five TTA methods and five backbone architectures show that CoDA consistently improves independent data–model adaptation. The gains persist across diverse evaluation protocols and further extend to natural rendition shifts on ImageNet-R and geometry-aware corruptions on ImageNet-3DCC.

2 Related Work

Unlike training-time domain generalization methods based on adversarial or randomized transformations (Xu et al., 2021; Gokhale et al., 2023), test-time adaptation adjusts a source-trained classifier on unlabeled target data at inference time, without access to source data or target labels (Chi et al.,

2021; Gao et al., 2022; Jain & Learned-Miller, 2011; Jang et al., 2023; Liu et al., 2021; Liang et al., 2020). Under corruption shifts, existing methods mainly follow two routes, focusing on either model adaptation or data adaptation that transforms target inputs before prediction, while recent work has also begun to connect these two directions through diffusion-based adaptation.

Model Adaptation. Model-adaptation methods update either the model parameters or the prediction rule using unlabeled test inputs. Some methods make lightweight changes in prediction space, for example by adapting classifier weights, prototypes, or prediction structure online (Iwasawa & Matsuo, 2021; Boudiaf et al., 2022; Su et al., 2022; Chen et al., 2022). AdaBN recalibrates normalization statistics, and Tent adapts normalization parameters by entropy minimization at test time (Schneider et al., 2020; Wang et al., 2020; Lim et al., 2023; Zhao et al., 2023). Another group uses auxiliary objectives, including self-supervised losses (Sun et al., 2020; Gandelsman et al., 2022) and consistency regularization across augmented views or teacher-student predictions (Zhang et al., 2022; Wang et al., 2022; Döbler et al., 2023). EATA, SAR, DeYO, COME, and ZeroSiam focus on making updates more reliable by filtering samples or using more conservative objectives (Niu et al., 2022; 2023; Lee et al., 2024; Zhang et al., 2025; Chen et al., 2026). In continual or non-stationary test streams, NOTE, Cotta, RoTTA, and SoTTA further stabilize adaptation with memory, normalization, or teacher-student mechanisms (Gong et al., 2022; Wang et al., 2022; Yuan et al., 2023; Gong et al., 2023). Although these TTA methods differ in objective and update scope, they typically update the classifier from a single corrupted view, or from augmentations derived from that view. Our setting differs in that model updates are conditioned on agreement between a corrupted observation and a restored prior.

Diffusion-based Adaptation. Diffusion models provide another route for test-time adaptation by either transforming target inputs or providing additional signals for model optimization. Some diffusion-based methods use diffusion models as discriminators or as guidance signals during test-time optimization (Li et al., 2023; Prabhudesai et al., 2023; Li et al., 2024). More closely related to our setting are restoration-based methods, which use a pretrained diffusion model to map corrupted inputs toward a source-like or synthetic domain before prediction (Kawar et al., 2022; Wang et al., 2023; Nichol & Dhariwal, 2021; Nie et al., 2022; Gao et al., 2023; Tsai et al., 2024; Guo et al., 2025). DDA (Gao et al., 2023) uses reverse diffusion to restore target images for TTA. GDA (Tsai et al., 2024) adds structural, content, and style guidance to preserve semantics during restoration. SDA (Guo et al., 2025) further aligns the source model and target samples in the synthetic domain. And recent work (Yang et al., 2026) uses diffusion-calibrated instances to construct reliable supervision for a single adaptive model, without adapting a separate model to the synthetic domain. CMDA (Zhang et al., 2026) uses model predictions to guide synthesis and adapts the model on the resulting synthetic samples, without a parallel adaptation path on the original corrupted inputs. In contrast, CoDA adapts both corrupted and restored views in parallel and explicitly controls their interaction.

3 PRELIMINARIES

We consider test-time adaptation for image classification under corruption shifts. Let $f_{\theta_0} = h \circ g_{\theta_0}$ denote a classifier trained on a labeled source domain $\mathcal{D}_s$, where g_{θ_0} is the feature extractor and h is the classification head. At test time, source data are unavailable, and the model receives an unlabeled target stream $\mathcal{D}_t = \{x_i\}_{i=1}^N$. TTA updates the model using batches from $\mathcal{D}_t$, without access to target labels. For each target sample, we construct an observation–restoration pair (x_i^o, x_i^r). The corrupted image is treated as the observation:

$$x_i^o = x_i. \tag{1}$$

A diffusion restoration model $R(\cdot)$ maps the observation to a restored view:

$$x_i^r = R(x_i^o). \tag{2}$$

The restored view is often closer to the source distribution in appearance, but it is not an independent measurement of the test sample. It is a model-based estimate shaped by a learned image prior and may smooth discriminative details or introduce plausible but incorrect cues. We therefore treat x_i^r as a prior-informed restored estimate rather than a replacement for x_i^o. Under the default DDA-based setting, we maintain two adaptation branches initialized from the same source classifier:

$$\theta^o, \theta^r \leftarrow \theta_0. \tag{3}$$

Their logits and predictive distributions are

$$z_i^o = h(g_{\theta^o}(x_i^o)), \quad z_i^r = h(g_{\theta^r}(x_i^r)), \tag{4}$$

$$p_i^o = \mathrm{softmax}(z_i^o), \quad p_i^r = \mathrm{softmax}(z_i^r). \tag{5}$$

4 METHOD

Overview. As shown in Figure 1, motivated by the data-assimilation view of combining observation and prior-informed information, CoDA regulates cross-view interaction through three stages. **Cross-View Gating** filters branch updates according to cross-view compatibility (Sec. 4.1). **Selective Coupling** limits cross-branch drift by applying consistency only to samples that pass both gates and agree at top-1 (Sec. 4.2). Finally, **Entropy-Aware Prediction Fusion** combines the two branch logits using entropy-based weights (Sec. 4.3). The underlying branch-specific TTA objective is retained.

4.1 CROSS-VIEW GATING

Motivated by the data-assimilation principle of reconciling observation and prior-informed information, CoDA uses the complementary view to validate branch updates. We operationalize this validation through cross-view compatibility: the top-1 prediction of one branch should remain among the top-k hypotheses of the complementary branch. In this way, cross-view support provides an additional criterion for controlling branch updates.

We build this gate on top of the sample selection rule of the underlying TTA method. Let $\mathcal{S}_{\mathrm{tta}}^b$ denote the candidate update set selected for branch $b \in \{o, r\}$ by the base TTA method, where o and r denote the observation and prior branches. For each sample x_i, the two branches produce predictive distributions p_i^o and p_i^r. For branch b, we denote its top-1 class by $\hat{y}_i^b$ and the top-k set of the opposite branch $\bar{b}$ by $\mathcal{Y}_{i,k}^{\bar{b}}$:

$$\hat{y}_i^b = \arg\max_y p_i^b(y), \qquad \mathcal{Y}_{i,k}^{\bar{b}} = \mathrm{TopK}\left(p_i^{\bar{b}}\right). \tag{6}$$

For the prior branch, cross-view gating is always applied. For the observation branch, the top-k gate is activated only when the restored-prior prediction has lower entropy, $H(p_i^r) < H(p_i^o)$. This asymmetric gating avoids over-constraining the observation branch when the restored-prior prediction provides weaker confidence support. When cross-view gating is active, the resulting update mask is

$$m_i^b = \mathbb{I}\left[i \in \mathcal{S}_{\mathrm{tta}}^b\right] \cdot \mathbb{I}\left[\hat{y}_i^b \in \mathcal{Y}_{i,k}^{\bar{b}}\right]. \tag{7}$$

Given this mask, the branch loss is:

$$\mathcal{L}_{\mathrm{branch}}^b = \frac{\sum_{i \in \mathcal{D}_t} m_i^b \mathcal{L}_{\mathrm{tta}}\left(p_i^b\right)}{\sum_{i \in \mathcal{D}_t} m_i^b}, \qquad b \in \{o, r\}, \tag{8}$$

where $\mathcal{L}_{\mathrm{tta}}$ is the objective of the underlying TTA method. Overall, Cross-View Gating regulates model adaptation through cross-view compatibility, enabling conditional interaction between the two branches while preserving the underlying TTA objective.

4.2 SELECTIVE COUPLING WITH JENSEN-SHANNON DIVERGENCE

Cross-View Gating removes unsupported updates, but the two branches can still drift apart as they adapt to different views. Persistent independent drift may gradually weaken the relation between their predictions, making cross-view guidance less stable and prediction fusion less reliable. To mitigate this issue, we add a selective coupling step that aligns the two branches on samples that pass both gates and already agree at top-1. Given the branch masks m_i^o and m_i^r from Cross-View Gating, we define a consensus mask:

$$m_i^{\mathrm{con}} = m_i^o \cdot m_i^r \cdot \mathbb{I}\left[\hat{y}_i^o = \hat{y}_i^r\right]. \tag{9}$$

This mask selects samples that pass both gates and share the same top-1 class. For each selected sample, we compute the mean predictive distribution:

$$\bar{p}_i = \frac{1}{2}\left(p_i^o + p_i^r\right). \tag{10}$$

We then use the symmetric Jensen-Shannon divergence to couple the two predictions without assigning fixed teacher-student roles:

$$\mathcal{L}_{\mathrm{JS}}\left(p_i^o, p_i^r\right) = \frac{1}{2} D_{\mathrm{KL}}\left(p_i^o \| \bar{p}_i\right) + \frac{1}{2} D_{\mathrm{KL}}\left(p_i^r \| \bar{p}_i\right). \tag{11}$$

The consensus loss is:

$$\mathcal{L}_{\mathrm{con}} = \frac{\sum_{i \in \mathcal{D}_t} m_i^{\mathrm{con}} \mathcal{L}_{\mathrm{JS}}\left(p_i^o, p_i^r\right)}{\sum_{i \in \mathcal{D}_t} m_i^{\mathrm{con}}}. \tag{12}$$

By coupling only consensus samples, Selective Coupling avoids introducing additional bias from aligning conflicting predictions, while reducing potential cross-branch drift and preserving the complementary predictive roles of the two branches.

4.3 Entropy-Aware Prediction Fusion

After adaptation, CoDA combines the two branch predictions with sample-dependent weights. Following the data-assimilation intuition of weighted information reconciliation, we use predictive entropy to determine their relative contributions. For the observation branch, we define:

$$\rho_i^o = 1 - H(p_i^o), \qquad H(p_i) = -\frac{1}{\log C} \sum_{c=1}^{C} p_i(c) \log p_i(c), \tag{13}$$

where C is the number of classes. For the prior branch, we further account for the prediction change induced by restoration. Specifically, we evaluate the corrupted observation using the same prior branch:

$$p_i^{o \to r} = \mathrm{softmax}\left(h(g_{\theta^r}(x_i^o))\right), \tag{14}$$

and define the entropy ratio to quantify the change relative to the restored view:

$$\mathrm{rel}_i^r = \frac{H_i^{o \to r}}{H_i^r}. \tag{15}$$

We use this change to further adjust the entropy-based contribution of the prior branch: a decrease in entropy strengthens its contribution, whereas an increase in entropy further suppresses it. To stabilize this adjustment across target batches, we maintain an exponential moving average with initial value $R_0^r = 1$:

$$R_t^r = \beta R_{t-1}^r + (1 - \beta)\mathrm{rel}_t^r, \tag{16}$$

where rel_t^r is the batch average of rel_i^r at test step t, and β is the momentum coefficient. Unlike ROID (Marsden et al., 2024), which uses EMA for class-wise prediction estimation, our EMA tracks the batch-averaged entropy ratio. And the resulting prior-branch weight is

$$\rho_i^r = R_t^r \left(1 - H(p_i^r)\right). \tag{17}$$

Finally, CoDA combines the two branch logits with the normalized weights:

$$\hat{y}_i = \arg\max_y \mathrm{softmax}\left(\frac{\rho_i^o}{\rho_i^o + \rho_i^r} z_i^o + \frac{\rho_i^r}{\rho_i^o + \rho_i^r} z_i^r\right)_y. \tag{18}$$

The final prediction combines the two branch logits, with the prior-branch contribution adaptively adjusted by the entropy change induced by restoration.

5 Experiment

Datasets and metrics. We use ImageNet-C (Hendrycks & Dietterich, 2019) as the main benchmark and evaluate all corruption experiments at severity level 5. ImageNet-C contains 15 corruption types grouped into four categories: Noise (N), Blur (B), Weather (W), and Digital (D). We report top-1 accuracy (%), including category-wise averages and the average over all 15 corruptions. To evaluate generalization beyond standard 2D corruptions, we further consider ImageNet-R (Hendrycks et al., 2021), which contains natural rendition shifts, and ImageNet-3DCC (Kar et al., 2022), which introduces more realistic corruptions using scene geometry. Additional dataset details are provided in Appendix A.1.

Table 1: **Results on ImageNet-C under the Regular episodic protocol.** We report top-1 accuracy (%) over four corruption categories Noise (N), Blur (B), Weather (W), and Digital (D), with Avg. averaged over all 15 corruption types.

Method	ResNet50-GN				ViT-Base				ViT-Small				ConvNeXt-Tiny				Swin-Tiny				Avg.
	$N\uparrow$	$B\uparrow$	$W\uparrow$	$D\uparrow$	$N\uparrow$	$B\uparrow$	$W\uparrow$	$D\uparrow$	$N\uparrow$	$B\uparrow$	$W\uparrow$	$D\uparrow$	$N\uparrow$	$B\uparrow$	$W\uparrow$	$D\uparrow$	$N\uparrow$	$B\uparrow$	$W\uparrow$	$D\uparrow$	
No Adapt	18.6	19.4	47.7	33.9	8.2	28.4	36.1	41.6	1.9	23.4	27.7	33.5	23.8	24.7	52.1	35.4	22.0	20.5	48.0	24.0	29.4
DDA	50.4	21.8	40.8	42.4	50.7	29.3	35.4	46.9	46.6	24.3	27.6	40.8	52.7	27.2	44.0	45.4	46.9	22.1	39.0	36.3	37.8
DDA+EATA	54.9	33.1	46.1	49.9	62.4	57.3	64.6	66.8	53.1	**45.9**	53.4	58.1	55.7	**33.7**	50.7	51.8	53.9	36.7	49.9	51.2	51.1
CoDA+EATA	**55.4**	**34.2**	**53.6**	**53.5**	**62.7**	56.6	**66.0**	**67.3**	**55.2**	45.5	**55.0**	**59.1**	**55.8**	33.6	**53.2**	**53.6**	**54.2**	**36.9**	**52.0**	**52.7**	**52.6**
DDA+SAR	53.7	26.0	39.7	42.7	57.3	52.2	56.2	62.0	42.5	36.7	40.7	52.4	55.2	29.3	43.4	50.3	50.6	26.7	40.3	40.2	44.4
CoDA+SAR	**54.2**	**26.4**	**40.0**	**48.2**	**59.8**	**52.7**	**61.4**	**63.9**	**49.2**	**40.0**	**46.2**	**54.5**	**55.3**	**29.3**	**44.4**	**52.0**	**51.3**	**26.8**	**40.6**	**45.7**	**46.6**
DDA+DeYO	56.2	29.8	42.7	47.7	63.0	**56.2**	57.8	62.7	17.2	38.9	41.0	55.4	**55.9**	**25.9**	42.3	39.5	52.9	**27.7**	40.1	41.6	44.4
CoDA+DeYO	**56.6**	**31.1**	**52.2**	**53.8**	**63.3**	55.9	**68.7**	**69.2**	**19.5**	**44.2**	**55.1**	**61.0**	**55.9**	24.7	**42.8**	**47.7**	**53.5**	26.8	**44.3**	**50.4**	**48.7**
DDA+COME	51.8	22.7	41.0	43.1	61.9	58.0	49.6	53.9	**23.8**	37.9	40.1	53.5	57.2	**29.9**	48.6	46.9	**55.9**	27.4	46.0	47.0	44.5
CoDA+COME	**52.7**	**23.1**	**41.5**	**44.9**	**62.0**	**58.1**	**69.2**	**68.9**	22.0	**43.2**	**53.4**	**61.1**	**57.3**	28.2	**52.3**	**52.7**	55.4	**29.2**	**54.8**	**49.7**	**48.9**
DDA+ZeroSiam	56.3	37.7	55.5	55.5	62.2	57.5	64.8	66.9	53.6	48.3	55.6	59.6	56.6	**37.1**	54.0	54.3	53.7	37.7	49.5	50.9	53.2
CoDA+ZeroSiam	**57.1**	**38.3**	**59.1**	**57.7**	**62.9**	**58.0**	**67.5**	**68.2**	**56.0**	**48.8**	**58.2**	**61.0**	**56.9**	36.6	**56.8**	**56.1**	**54.5**	**37.8**	**52.4**	**53.6**	**54.7**

Methods and models. We instantiate CoDA with EATA (Niu et al., 2022), SAR (Niu et al., 2023), DeYO (Lee et al., 2024), COME (Zhang et al., 2025), and ZeroSiam (Chen et al., 2026). Our default restoration pipeline is DDA (Gao et al., 2023), while SDA (Guo et al., 2025) and GDA (Tsai et al., 2024) are evaluated separately. For each TTA method, *DDA+TTA* independently adapts two source-initialized models to the corrupted and restored views, whereas *CoDA+TTA* couples them via our CoDA protocol. For SDA, the restored-prior branch uses the SDA-trained model, whereas the observation branch uses the original source model. We evaluate ResNet50-GN, ViT-Base, ViT-Small, ConvNeXt-Tiny, and Swin-Tiny from `timm` (Wightman, 2019). Unless stated otherwise, each base TTA method follows its original configuration, and Cross-View Gating uses $k = 5$. Further implementation details are provided in Appendix A.2.

Evaluation protocols. Three evaluation protocols are considered on ImageNet-C. *Regular episodic*: samples are randomly shuffled within each corruption, and the model is reset before each corruption. *Label shift*: based on the extreme class-imbalance setting studied in prior TTA work (Niu et al., 2023; Lee et al., 2024; Chen et al., 2026), test stream with Imbalance=∞ is constructed. *Continual*: the 15 corruption domains are processed sequentially without resetting the adapted model, with samples randomly shuffled within each corruption (Wang et al., 2022). All experiments are conducted on NVIDIA RTX 4090 GPU.

5.1 Controlled Co-Adaptation Improves Data–Model Adaptation

Simply combining data and model adaptation is not sufficient. Figure 2 compares four adaptation strategies across five TTA methods. Data adaptation improves the source model, confirming that restoration provides useful information, and combining it with model adaptation yields further gains. However, independent data-model adaptation does not consistently outperform model adaptation alone. CoDA instead gives the strongest overall performance across all five TTA methods, indicating the importance of controlled cross-view interaction.

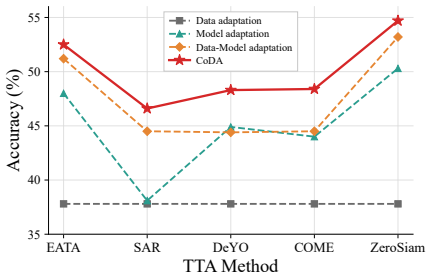


Figure 2: Comparison of different adaptation strategies on ImageNet-C.

Controlled interaction provides consistent additional gains. Table 1 provides a systematic comparison across five TTA methods and five backbones. DDA alone improves average accuracy from 29.4% to 37.8%, while independent data–model adaptation further raises performance to 44.4–53.2%, demonstrating the benefit of jointly exploiting the two adaptation routes. On top of these strong baselines, CoDA improves all five DDA+TTA variants by 2.8 points on average and up to 4.4 points, reaching 54.7% with ZeroSiam.

CoDA remains effective with SDA-aligned models. The benefit of CoDA is not limited to the default DDA setting. Unlike DDA, SDA additionally adapts the prior-branch classifier to the restored

Table 2: **SDA-based results on ImageNet-C under the Regular episodic protocol.** Each entry reports **SDA+TTA / CoDA+TTA** top-1 accuracy (%), averaged over all 15 corruption types, with CoDA built upon the SDA setting

Method	ResNet50-GN	ViT-B	ViT-S	Conv-T	Swin-T	Avg.
SDA	40.6	43.5	37.8	42.0	35.1	39.8
+EATA	49.4/**50.3**	63.8/**64.7**	55.4/**56.5**	47.1/**48.0**	47.6/**48.3**	52.7/**53.5**
+SAR	43.6/**44.6**	59.1/**60.1**	47.4/**49.2**	44.2/**45.0**	38.5/**39.9**	46.5/**47.8**
+DeYO	48.4/**49.1**	63.1/**64.0**	44.7/**46.3**	42.0/**43.1**	43.3/**44.0**	48.3/**49.3**
+COME	41.8/**42.9**	63.4/**64.1**	44.4/**45.5**	46.8/**47.3**	46.3/**46.7**	48.5/**49.3**
+ZeroSiam	53.7/**55.1**	62.7/**64.4**	55.7/**57.4**	50.1/**51.5**	47.3/**49.9**	53.9/**55.6**

Table 3: Runtime and accuracy on ImageNet-C. CoDA is instantiated with SAR. DPM denotes DPM-Solver++

Method	Time (s)	ViT-S	ConvT
DDA	14.6	34.0	41.6
+DPM	2.0	31.3	38.7
SDA	14.6	37.8	42.0
+DPM	2.0	35.4	39.2
CoDA	14.7	**47.2**	**46.0**
+DPM	2.1	42.9	42.6

synthetic domain before test-time adaptation. Even with this restoration-aware initialization, CoDA improves all five TTA methods on every evaluated backbone (Table 2), by 1.1 points on average and up to 1.7 points. With ZeroSiam, the average accuracy further increases from 53.9% to 55.6%.

Runtime comparison with restoration-based adaptation. As shown in Table 3, CoDA takes 14.7 s compared with 14.6 s for DDA/SDA, while improving accuracy by 7.8 points on average. With DPM-Solver++, the corresponding runtimes are 2.1 s and 2.0 s, with an average gain of 6.6 points. Overall, CoDA provides consistent accuracy gains over restoration-based data adaptation with only a small increase in runtime. Appendix C.1 further shows that replacing the default diffusion restorer with Restormer (Zamir et al., 2022) and AdaIR (Cui et al., 2025) still yields consistent improvements, suggesting that CoDA is not tied to a specific restoration architecture.

5.2 Why Controlled Cross-View Interaction Matters

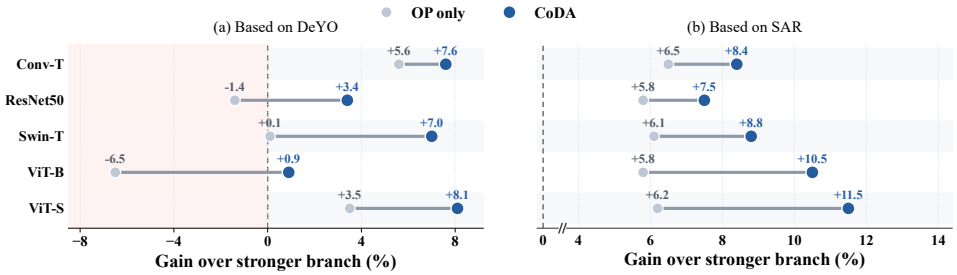


Figure 3: **CoDA remains effective at batch size one.** We report the gains of direct observation–prior fusion (OP only) and full CoDA over the stronger of the observation and prior branches. CoDA consistently improves the stronger branch across backbones.

Direct cross-view fusion can be unstable. Figure 3 reports the gain over the stronger branch under the label shift setting with batch size one. Direct fusion (OP only) improves some backbones, but its gains are inconsistent, dropping to -6.5% on ViT-B and -1.4% on ResNet50 with DeYO. In contrast, CoDA consistently improves the stronger branch across all backbones, with gains of 0.9%–8.1% for DeYO and 7.5%–11.5% for SAR. This contrast motivates explicit control over cross-view interaction rather than unconditional fusion.

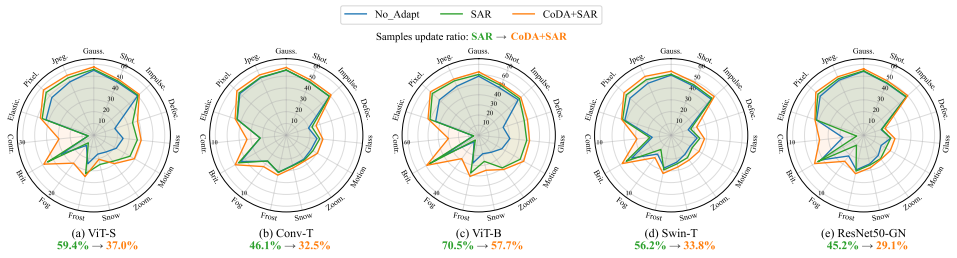


Figure 4: Cross-View Gating improves prior-branch adaptation with fewer updates. Across five backbones, CoDA+SAR improves or maintains accuracy while substantially reducing the sample update ratio compared with direct SAR adaptation on restored images.

Cross-view gating enables stronger adaptation with fewer updates. The restored view can be visually cleaner yet semantically biased, making confidence alone insufficient for selecting updates. As shown in Figure 4, Cross-View Gating uses substantially fewer samples while improving or preserving prior-branch accuracy across all five backbones. Averaged over backbones, the update ratio decreases from 55.5% to 38.0%, while accuracy increases from 45.3% to 46.3%. The gain therefore comes from more reliable updates rather than broader update coverage. Detailed backbone-wise accuracy and update-ratio results are provided in Appendix C.2.

Each interaction control contributes to CoDA. Table 4 separates the benefit of dual-branch adaptation from that of the three interaction controls. Introducing the observation–prior (O-P) branches alone improves accuracy from 34.06% to 39.66% on ResNet50 and from 54.67% to 56.89% on ViT-B, confirming that jointly adapting the two views is already beneficial. Building on this baseline, Cross-View Gating (CVG), Selective Coupling (SC), and Entropy-aware Prediction Fusion (EAF) each further improve performance on both backbones. Using all three components yields the best results, reaching 41.40% on ResNet50 and 59.41% on ViT-B, corresponding to gains of 1.74% and 2.52% over the uncontrolled O-P baseline. Appendix D.1 provides further analysis of k and entropy-aware fusion.

Table 4: **CoDA component ablation with SAR as the base TTA method.**

O-P	CVG	SC	EAF	ResNet50	ViT-B
–	–	–	–	34.06	54.67
✓	–	–	–	39.66	56.89
✓	✓	–	–	40.83	58.39
✓	–	✓	–	40.32	57.86
✓	–	–	✓	40.60	57.60
✓	✓	✓	✓	**41.40**	**59.41**

5.3 Robustness Across Evaluation Protocols

Table 5: **Comparisons with SOTAs on ImageNet-C under the label-shifted protocol with batch size 1.** We report top-1 accuracy (%) over N/B/W/D and the average over all 15 corruption types.

Method	ResNet50-GN				ViT-Base				ViT-Small				ConvNeXt-Tiny				Swin-Tiny				Avg.
	$N\uparrow$	$B\uparrow$	$W\uparrow$	$D\uparrow$	$N\uparrow$	$B\uparrow$	$W\uparrow$	$D\uparrow$	$N\uparrow$	$B\uparrow$	$W\uparrow$	$D\uparrow$	$N\uparrow$	$B\uparrow$	$W\uparrow$	$D\uparrow$	$N\uparrow$	$B\uparrow$	$W\uparrow$	$D\uparrow$	
DDA+EATA	50.8	22.3	39.9	43.9	52.0	34.2	40.4	50.8	46.7	26.2	29.4	43.3	53.0	27.4	44.1	46.4	47.7	22.6	38.9	39.0	39.3
CoDA+EATA	**52.4**	**23.5**	**42.8**	**46.4**	**55.4**	**35.1**	**42.1**	**53.3**	**53.0**	**27.5**	**32.4**	**45.9**	**54.0**	**28.7**	**46.8**	**49.1**	**49.8**	**24.7**	**41.6**	**43.7**	**41.7**
DDA+SAR	52.0	25.2	43.5	45.4	53.5	41.0	46.0	52.9	46.7	33.1	36.8	45.4	53.6	28.1	44.2	47.2	48.7	24.5	39.6	41.5	41.9
CoDA+SAR	**53.3**	**26.4**	**44.6**	**48.5**	**56.7**	**45.2**	**52.5**	**57.3**	**53.0**	**38.7**	**41.7**	**50.1**	**54.6**	**29.4**	**46.8**	**49.7**	**50.9**	**26.9**	**42.1**	**45.1**	**45.1**
DDA+DeYO	55.9	27.0	47.8	52.8	61.5	44.8	54.9	60.9	32.2	**40.7**	52.6	58.2	54.5	**26.2**	49.2	49.5	51.0	22.1	42.7	43.0	46.1
CoDA+DeYO	**57.3**	**27.1**	**58.9**	**58.6**	**63.6**	**47.2**	**69.7**	**70.0**	**41.6**	39.9	**59.2**	**62.4**	**55.7**	24.3	**56.3**	**50.8**	**53.3**	**27.1**	**51.8**	**53.3**	**51.2**
DDA+COME	50.3	21.8	40.5	41.9	60.0	43.6	47.5	53.6	34.9	38.6	45.4	53.5	56.3	**32.3**	54.4	56.5	54.5	21.4	45.7	46.7	44.6
CoDA+COME	**52.6**	**23.1**	**41.1**	**44.9**	**62.2**	**45.8**	**68.3**	**68.8**	**46.7**	**38.9**	**57.1**	**61.5**	**57.3**	31.5	**60.4**	**57.8**	**55.0**	**30.1**	**55.4**	**54.5**	**50.4**
DDA+ZeroSiam	54.7	38.6	**57.7**	**57.7**	61.2	56.4	62.8	64.3	51.6	45.6	52.9	57.0	56.1	38.6	54.3	54.6	53.0	36.8	48.8	49.6	52.4
CoDA+ZeroSiam	**56.3**	**39.5**	56.4	57.1	**63.1**	**58.0**	**67.1**	**67.8**	**55.5**	**47.1**	**56.8**	**59.9**	**56.9**	**39.2**	**57.3**	**57.1**	**54.9**	**38.9**	**53.4**	**54.0**	**54.6**

Stable gains under label shift with batch size one. We further consider an extremely label-imbalanced test stream with batch size one, where each update is computed from a single target sample. Under this setting, CoDA improves all five DDA+TTA baselines by 3.7 points on average and up to 5.8 points (Table 5). COME improves from 44.6% to 50.4%, while CoDA+ZeroSiam reaches the highest average accuracy of 54.6%. The gains persist without relying on multi-sample adaptation batches. For completeness, Appendix B.2 and B.3 provide additional per-corruption results under label-shifted streams with batch sizes 64 and 1.

Table 6: **Continual adaptation results on ImageNet-C.** We report per-corruption and average top-1 accuracy (%) over the 15 corruption types. Processed sequentially from left to right in the table.

Model+Method	t →															Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
ResNet-50-BN	42.6	38.0	37.4	16.8	17.6	15.6	23.8	21.9	30.3	18.3	**54.7**	3.1	37.4	43.6	**48.8**	30.0
DDA+RoTTA	**52.4**	50.2	50.1	**26.4**	**33.1**	30.5	32.8	29.3	31.5	20.1	41.5	14.1	37.8	38.2	37.9	35.1
CoDA+RoTTA	52.3	**50.3**	**51.1**	25.7	32.7	**31.3**	**34.2**	**30.9**	**33.8**	**20.4**	45.3	14.1	**42.4**	**43.9**	43.5	**36.8**
ConvNeXt-T	21.5	27.7	22.2	24.6	10.9	32.5	30.9	44.4	52.5	39.8	72.0	44.7	23.2	17.6	55.7	34.7
DDA+EATA	56.8	54.0	53.8	30.1	35.2	36.6	**39.0**	41.8	51.4	42.9	66.9	34.5	46.7	54.9	62.0	47.1
CoDA+EATA	**58.8**	**54.7**	**54.7**	**31.6**	**36.5**	**38.5**	38.9	**43.7**	**53.8**	**49.1**	**69.2**	**43.2**	**47.6**	**59.2**	**63.4**	**49.5**
DDA+SAR	56.8	53.4	53.3	25.3	28.0	33.1	32.4	38.7	48.1	31.6	65.4	24.8	37.0	43.4	60.7	42.1
CoDA+SAR	**58.6**	**54.3**	**54.3**	**26.7**	**31.4**	**34.7**	**32.8**	**40.4**	**50.4**	**45.2**	**68.2**	**37.3**	**43.7**	**56.7**	**61.7**	**46.4**
DDA+ZeroSiam	58.1	56.1	55.8	28.3	35.2	36.4	37.0	40.3	50.3	36.9	66.5	23.7	45.8	55.3	61.9	45.8
CoDA+ZeroSiam	**59.8**	**56.7**	**57.1**	**29.2**	**36.6**	**37.6**	**38.1**	**42.2**	**53.0**	**44.9**	**68.1**	**42.5**	**47.7**	**60.1**	**63.2**	**49.1**

Table 8: **Generalization to 3D corruption shifts on ImageNet-3DCC.** We report top-1 accuracy (%) across five backbone architectures and their average.

Method	ResNet50	ViT-B	ViT-S	Conv-T	Swin-T	Avg.
Source	39.82	31.55	26.92	43.73	38.61	36.13
EATA	47.86	59.60	48.46	48.43	46.59	50.19
+DDA/CoDA	46.83/**48.61**	58.92/**60.10**	49.94/**49.95**	48.38/**48.85**	47.58/**47.70**	50.33/**51.04**
SAR	45.05	54.79	41.92	47.14	41.79	46.14
+DDA/CoDA	45.47/**46.18**	54.50/**55.30**	44.88/**46.25**	47.28/**47.74**	43.62/**43.92**	47.15/**47.88**
DeYO	47.88	57.47	47.77	47.21	43.03	48.67
+DDA/CoDA	47.77/**48.63**	56.69/**58.49**	49.61/**50.04**	47.69/**48.00**	44.30/**44.60**	49.21/**49.95**
ZeroSiam	50.04	59.48	49.85	50.33	46.36	51.21
+DDA/CoDA	50.05/**51.01**	58.79/**59.58**	51.19/**52.06**	49.98/**50.83**	47.23/**47.96**	51.45/**52.29**

Cross-domain continual adaptation. In continual adaptation, model states are retained as the 15 corruption domains are encountered sequentially. CoDA consistently improves the corresponding DDA+TTA baselines, with an average gain of 2.9% and up to 4.3% (Table 6). The largest improvement is obtained with SAR on ConvNeXt-T, from 42.1% to 46.4%, while CoDA also improves RoTTA on ResNet50-BN from 35.1% to 36.8%. The positive gains show that the co-adaptation protocol remains useful during prolonged adaptation, not only within isolated corruption domains.

5.4 Generalization Beyond ImageNet-C

Natural rendition shifts. The benefit also extends beyond ImageNet-C corruptions. On ImageNet-R (Table 7), ZeroSiam alone outperforms GDA, and independent GDA+ZeroSiam further improves both backbones. CoDA raises this independent baseline from 46.0% to 47.4% on ResNet50-GN and from 58.1% to 59.2% on ViT-B. These results suggest that CoDA can provide further gains when both adaptation routes are effective under a natural rendition shift.

Table 7: **Generalization to natural rendition shifts on ImageNet-R.** We report top-1 accuracy (%) on ResNet50-GN and ViT-Base.

Method	ResNet50	ViT-B
Source	40.8	43.1
GDA	42.7	44.7
ZeroSiam	43.7	56.2
GDA+ZeroSiam	46.0	58.1
CoDA+ZeroSiam	**47.4**	**59.2**

Geometry-aware corruptions. ImageNet-3DCC introduces geometry-aware corruption shifts that differ substantially from standard 2D corruptions, making restoration less consistently effective. Indeed, directly adding DDA to model adaptation provides only limited benefits and can even degrade performance on some backbones. In contrast, CoDA improves the corresponding DDA+TTA results across all 20 method–backbone combinations (Table 8). These results show that CoDA can turn the inconsistent contribution of restoration into consistent improvements by regulating how the restored prior interacts with the corrupted observation.

Additional evaluations. We further assess CoDA on CIFAR-10-C (Appendix B.4) and examine source-domain preservation on ImageNet (Appendix D.3). The latter considers both direct application without distribution shift and source-domain evaluation after corruption adaptation, showing that CoDA largely preserves source-model performance in both settings.

6 Conclusion

We presented CoDA, a test-time co-adaptation framework for jointly adapting to corrupted observations and their restored prior-informed views. Motivated by data assimilation, CoDA regulates cross-view interaction through cross-view gating, selective Jensen–Shannon coupling, and entropy-aware prediction fusion, while leaving the underlying TTA objective unchanged. Across ImageNet-C, CoDA consistently improves independent data–model adaptation over multiple TTA methods and backbone architectures, with gains persisting under challenging settings including batch-size-one label shift and continual adaptation. The same controlled interaction further extends beyond standard 2D corruptions to natural rendition shifts on ImageNet-R and geometry-aware corruptions on ImageNet-3DCC. These results suggest that effectively combining data and model adaptation requires not merely exploiting both views, but controlling when and how they interact.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing, organization, the presentation of experimental results, and coding assistance for parts of the experimental implementation. All experiments were conducted by the authors, and all AI-assisted text, code, and interpretations were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper and the correctness of the experimental implementation.

REFERENCES

Gustavo A Vargas Hakim, David Osowiechi, Mehrdad Noori, Milad Cheraghalikhani, Ali Bahri, Moslem Yazdanpanah, Ismail Ben Ayed, and Christian Desrosiers. Clipartt: Adaptation of clip to new domains at test time. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 7092–7101, February 2025.

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

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 295–305, 2022.

Guohao Chen, Shuaicheng Niu, Deyu Chen, Jiahao Yang, Zitian Zhang, Mingkui Tan, Pengcheng Wu, and Zhiqi Shen. Zerosiam: An efficient asymmetry for test-time entropy optimization without collapse. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9137–9146, 2021.

Hamadi Chihaoui, Abdelhak Lemkhenter, and Paolo Favaro. Blind image restoration via fast diffusion inversion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 34513–34532. Curran Associates, Inc., 2024. doi: 10.52202/079017-1088.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Yuning Cui, Syed Waqas Zamir, Salman Khan, Alois Knoll, Mubarak Shah, and Fahad Shahbaz Khan. AdaIR: Adaptive all-in-one image restoration via frequency mining and modulation. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Mario Döbler, Robert A Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7704–7714, 2023.

Geir Evensen. *Data assimilation: the ensemble Kalman filter*. Springer, 2009.

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

Jin Gao, Jialing Zhang, Xihui Liu, Trevor Darrell, Evan Shelhamer, and Dequan Wang. Back to the source: Diffusion-driven adaptation to test-time corruption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11786–11796, 2023.

Yunhe Gao, Xingjian Shi, Yi Zhu, Hao Wang, Zhiqiang Tang, Xiong Zhou, Mu Li, and Dimitris N Metaxas. Visual prompt tuning for test-time domain adaptation. *arXiv preprint arXiv:2210.04831*, 2022.

Tejas Gokhale, Rushil Anirudh, Jayaraman J. Thiagarajan, Bhavya Kailkhura, Chitta Baral, and Yezhou Yang. Improving diversity with adversarially learned transformations for domain generalization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 434–443, 2023.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

Taesik Gong, Yewon Kim, Taeckyung Lee, Sorn Chottananurak, and Sung-Ju Lee. Sotta: Robust test-time adaptation on noisy data streams. *Advances in Neural Information Processing Systems*, 36:14070–14093, 2023.

Jiayi Guo, Junhao Zhao, Chaoqun Du, Yulin Wang, Chunjiang Ge, Zanlin Ni, Shiji Song, Humphrey Shi, and Gao Huang. Everything to the synthetic: Diffusion-driven test-time adaptation via synthetic-domain alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30503–30513, 2025.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8349, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Peter L Houtekamer and Herschel L Mitchell. Data assimilation using an ensemble kalman filter technique. *Monthly weather review*, 126(3):796–811, 1998.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

Vidit Jain and Erik Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR 2011*, pp. 577–584. IEEE, 2011.

Minguk Jang, Sae-Young Chung, and Hye Won Chung. Test-Time Adaptation via Self-Training with Nearest Neighbor Information. In *International Conference on Learning Representations*, 2023.

Eugenia Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge university press, 2003.

Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18963–18974, 2022.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23593–23606. Curran Associates, Inc., 2022.

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

Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2206–2217, 2023.

Mingjia Li, Shuang Li, Tongrui Su, Longhui Yuan, Jian Liang, and Wei Li. Exploring structured semantic priors underlying diffusion score for test-time adaptation. *Advances in Neural Information Processing Systems*, 37:12164–12194, 2024.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 6028–6039, 2020.

Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. Ttn: A domain-shift aware batch normalization in test-time adaptation. In *International Conference on Learning Representations, ICLR 2023*, 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? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

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

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning*, pp. 16805–16827. PMLR, 2022.

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. *arXiv preprint arXiv:2302.12400*, 2023.

Mihir Prabhudesai, Tsung-Wei Ke, Alex Li, Deepak Pathak, and Katerina Fragkiadaki. Diffusion-tta: Test-time adaptation of discriminative models via generative feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 17567–17583. Curran Associates, Inc., 2023.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in neural information processing systems*, 33:11539–11551, 2020.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yongyi Su, Xun Xu, and Kui Jia. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering. *Advances in Neural Information Processing Systems*, 35: 17543–17555, 2022.

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

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

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7201–7211, 2022.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *International Conference on Learning Representations*, 2021.

Xu Yang, Moqi Li, and Kun Wei. Diffusion-calibrated continual test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27684–27692, 2026. doi: 10.1609/aaai.v40i33.39989.

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

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.

Chunyun Zhang, Fujun Yang, Chaoran Cui, Shuai Gong, Wenna Wang, Xue Lin, Yonggang Qi, and Lei Zhu. Collaborative model and data adaptation at test time. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.

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

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

Bowen Zhao, Chen Chen, and Shu-Tao Xia. Delta: Degradation-free fully test-time adaptation. In *International Conference on Learning Representations*, 2023.

Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 1219–1229, 2023.

A Additional Implementation Details

A.1 Additional Dataset Details

We evaluate CoDA on ImageNet-C (Hendrycks & Dietterich, 2019), a large-scale benchmark for common image corruptions. ImageNet-C is constructed by applying synthetic corruptions to the ImageNet validation set (Deng et al., 2009). Figure 5 shows examples of the corruption types. ImageNet-C contains 15 corruption types, including gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transformation, pixelation, and jpeg compression. Each corruption type is evaluated at five severity levels, where higher severity corresponds to a stronger distribution shift.

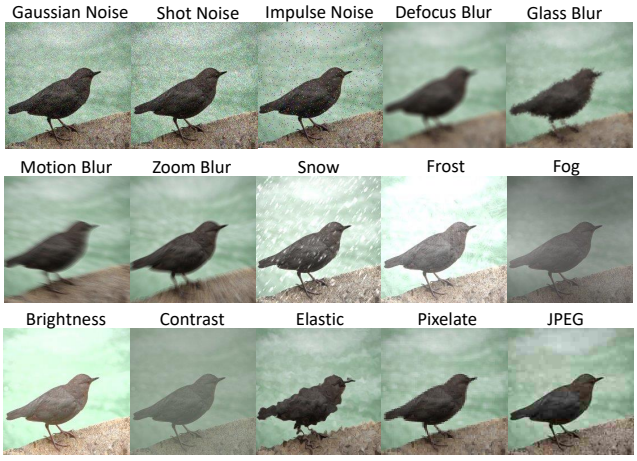


Figure 5: Examples of corruption types in ImageNet-C.

ImageNet-R. ImageNet-Rendition (Hendrycks et al., 2021) contains images from 200 ImageNet classes with diverse renditions, including paintings, cartoons, sketches, sculptures, and other visual styles. It evaluates robustness to natural rendition shifts that differ from the synthetic corruptions in ImageNet-C.

ImageNet-3DCC. ImageNet-3DCC (Kar et al., 2022) introduces corruptions that incorporate scene geometry, producing spatially structured degradations that reflect 3D scene properties. It provides a complementary evaluation beyond conventional 2D common corruptions.

CIFAR-10-C. CIFAR-10-C (Hendrycks & Dietterich, 2019) is a corruption benchmark constructed from the CIFAR-10 test set, covering common noise, blur, weather, and digital corruptions at five severity levels. We use it to evaluate CoDA on a smaller-scale corruption benchmark.

A.2 Implementation Details for CoDA and Baselines

All pretrained models used in this paper are publicly available. ResNet50-GN, ViT-Base, ViT-Small, ConvNeXt-Tiny, and Swin-Tiny are obtained from the `timm` repository (Wightman, 2019). WRN-28-10 is obtained from the official `AugMix` repository.

CoDA. The weight of the Jensen–Shannon term is fixed to 1. In Cross-View Gating, we use $k = 5$. In Entropy-Aware Prediction Fusion, we clip the EMA factor to $[0.5, 1.5]$. Unless otherwise specified, when CoDA is plugged into a baseline it uses the default hyperparameters of that baseline for the corresponding backbone.

A.2.1 Model-based Test-Time Adaptation Methods

SAR (Niu et al., 2023) We follow the hyperparameters from SAR whenever they are provided. We use SGD with momentum 0.9 and a batch size of 64, except in the batch-size-one experiments. The learning rates are $0.00025/0.001/0.0001/0.000025/0.00005$ for ResNet50-GN/ViT-Base/ViT-Small/ConvNeXt-Tiny/Swin-Tiny. For batch size 1, the learning rates are scaled 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 is set to $E_0 = 0.4 \ln C$, where C is the number of classes. The trainable parameters are the affine parameters of normalization layers from layer 1 to layer 3 in ResNet50-GN, from block 1 to block 8 in ViT-Base and ViT-Small, and all affine parameters in the other models.

EATA (Niu et al., 2022) Unless otherwise specified, we use the hyperparameters from EATA. In particular, we use the default setting of $2{,}000$ pre-collected in-distribution samples to compute Fisher importance.

DeYO (Lee et al., 2024) Unless otherwise specified, we use the hyperparameters from DeYO. The learning rates and trainable parameters follow SAR.

COME (Zhang et al., 2025) Following the experimental setup used for ZeroSiam, we implement COME on top of DeYO for comparison and use the same hyperparameters as DeYO.

ZeroSiam (Chen et al., 2026) Unless otherwise specified, we use the settings from ZeroSiam. In particular, `lr_scale`/`lr_p` are set to $10/100$, $5/5$, $10/100$, $10/100$, and $10/50$ for ResNet50-GN, ViT-Base, ViT-Small, ConvNeXt-Tiny, and Swin-Tiny, respectively.

RoTTA (Yuan et al., 2023) We use RoTTA for the continual adaptation experiment with ResNet-50-BN and follow its original implementation settings. The model is initialized from the source checkpoint and continuously adapted as the 15 ImageNet-C corruption domains are encountered sequentially, without resetting the model state between domains. For both DDA+RoTTA and CoDA+RoTTA, the RoTTA-specific memory, teacher-update, and optimization settings are kept unchanged. CoDA only modifies the interaction between the observation and restored-prior branches

A.2.2 Diffusion-based Data Adaptation Methods

DDA (Gao et al., 2023) For diffusion restoration, we use the same restoration pipeline and sampling configuration as DDA. Each corrupted input is first projected to an intermediate noisy state and then reconstructed by reverse diffusion. The number of forward diffusion steps is 50, the guidance scale is 6, and the low-frequency constraint scale D is 4. All other restoration hyperparameters follow the default DDA configuration. For fair comparison, the restored images are evaluated with the same backbone classifiers as in CoDA: ResNet50-GN, ViT-Base, ViT-Small, ConvNeXt-Tiny, and Swin-Tiny.

For the DPM-Solver++ (Lu et al., 2022) variant, the guidance scale is set to 8, the low-frequency constraint scale D remains 4, the number of forward noise-adding steps is 50, and the number of denoising steps is 5. All other restoration hyperparameters remain at their default values.

GDA (Tsai et al., 2024) We follow the original GDA implementation to construct the restored view in the ImageNet-R experiments. The same GDA configuration is used for GDA, GDA+ZeroSiam, and CoDA+ZeroSiam. In GDA+ZeroSiam, model adaptation is performed independently on the observation and GDA-restored views, whereas CoDA uses the same restored view while controlling cross-view interaction through the proposed co-adaptation protocol. All GDA-specific restoration settings are kept fixed across these comparisons.

SDA (Guo et al., 2025) For SDA, we follow the default implementation protocol. We use the same restoration parameters as in DDA: 50 forward noise-adding steps, guidance scale 6, and low-frequency constraint scale $D = 4$. All other restoration hyperparameters follow the default SDA configuration. SDA applies the same noise-adding and restoration process to images generated by Diffusion model and then trains the source-domain model on the restored synthetic images. Following the recommended setting of SDA, we generate $50{,}000$ images in total, with 50 images per class. The optimization settings and training schedule follow the original SDA protocol.

I-DDPM (Nichol & Dhariwal, 2021) For CIFAR-10-C image restoration, we use I-DDPM and follow the same general protocol as in DDA. Each corrupted image is first perturbed to a noisy state with 50 forward diffusion steps and then reconstructed through reverse diffusion. We apply a gradient-based correction based on the discrepancy between low-frequency components. In our implementation, the low-frequency constraint is set to 2 and its scale is set to 6. All other restoration hyperparameters follow the default settings.

B SUPPLEMENTARY RESULTS

This section provides additional results. We report model adaptation results that are not shown in the main paper and provide detailed results for all 15 ImageNet-C corruption types, while the main paper reports the Noise/Blur/Weather/Digital averages. All experimental settings follow Appendix A unless otherwise specified.

B.1 PER-CORRUPTION RESULTS UNDER THE REGULAR PROTOCOL

Table 9 reports per-corruption results on ImageNet-C at severity 5 under the regular episodic protocol. CoDA generally improves model adaptation across five backbones

Table 9: Per-corruption top-1 accuracy (%) on ImageNet-C at severity 5 under the regular episodic protocol. We report model adaptation methods and their CoDA versions across five backbones.

Model+Method	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
ResNet50-GN	18.0	19.8	17.9	19.8	11.3	21.4	24.9	40.4	47.3	33.6	69.3	36.3	18.6	28.4	52.3	30.6
EATA	37.7	40.6	38.6	28.5	27.5	36.8	38.4	49.7	47.3	54.6	72.0	49.0	41.7	55.2	56.9	45.0
CoDA+EATA	58.0	54.1	54.0	28.6	32.4	36.2	39.7	47.7	48.7	48.9	69.1	41.8	50.3	61.0	60.9	48.8
SAR	28.5	30.9	29.4	18.4	18.8	30.3	30.4	42.4	43.5	4.3	70.7	43.9	15.5	48.6	55.1	34.1
CoDA+SAR	57.1	52.9	52.6	18.2	25.8	29.9	31.8	42.9	45.9	3.8	67.3	36.2	40.1	57.7	58.8	41.4
DeYO	39.3	42.1	40.5	22.4	23.8	38.3	34.0	50.8	49.4	55.9	72.9	49.9	42.8	55.7	57.6	45.0
CoDA+DeYO	59.1	55.4	55.2	21.3	27.8	37.7	37.7	48.5	49.7	40.2	70.5	40.9	51.4	61.7	61.4	47.9
COME	17.9	19.7	17.7	19.7	11.3	21.4	24.9	40.4	47.5	33.6	69.2	35.9	18.6	27.9	52.1	30.5
CoDA+COME	56.1	51.2	50.9	19.5	23.9	22.5	26.4	39.4	44.6	18.2	63.7	27.2	42.2	52.9	57.1	39.7
ZeroSiam	40.0	42.8	41.5	31.9	32.2	41.1	44.5	53.6	53.2	59.0	72.8	51.8	48.2	58.6	59.0	48.7
CoDA+ZeroSiam	59.2	56.3	55.9	32.3	36.1	40.3	44.4	53.4	53.6	58.4	70.9	49.2	55.7	63.1	62.8	52.8
ViT-B	9.5	6.8	8.2	29.0	23.4	33.9	27.1	15.9	26.5	47.2	54.6	44.1	30.5	44.5	47.8	29.9
EATA	51.5	51.0	52.2	56.7	56.7	61.0	58.9	65.3	63.5	71.6	76.9	67.4	65.2	71.1	68.4	62.5
CoDA+EATA	65.3	61.0	61.6	54.4	56.8	58.9	56.4	60.8	60.5	68.5	74.1	64.2	64.5	71.6	68.8	63.2
SAR	44.5	17.3	45.6	53.2	50.0	55.9	51.3	57.6	53.0	66.5	74.8	64.4	55.4	66.6	64.1	54.7
CoDA+SAR	63.5	56.5	59.3	52.4	52.8	55.1	50.4	54.0	55.8	63.0	72.8	61.4	59.3	68.9	66.0	59.4
DeYO	52.8	53.0	54.0	58.0	58.5	63.0	49.8	67.2	65.6	73.3	78.0	68.1	67.7	73.1	70.2	63.5
CoDA+DeYO	65.9	61.5	62.5	57.5	59.0	61.9	45.1	64.0	63.9	70.6	76.4	65.2	67.4	73.5	70.7	64.3
COME	50.9	51.3	52.1	56.9	57.2	62.4	59.3	65.9	64.4	72.3	77.4	67.7	66.4	72.3	69.2	63.0
CoDA+COME	64.9	59.5	61.5	56.3	57.9	61.1	57.3	65.0	63.9	71.2	76.7	65.7	66.9	73.0	70.0	64.7
ZeroSiam	50.6	50.7	51.6	56.5	56.6	61.4	58.7	65.1	64.6	72.1	77.1	67.5	65.4	71.5	68.4	62.5
CoDA+ZeroSiam	65.3	61.5	61.8	56.1	57.8	60.3	57.7	62.3	63.0	69.3	75.5	64.6	66.2	72.3	69.6	64.2
ViT-S	2.0	2.1	1.5	24.1	17.2	30.3	22.1	9.5	19.3	37.8	44.4	30.0	24.8	38.2	41.1	23.0
EATA	22.4	22.9	23.8	45.8	44.3	48.5	44.4	49.7	50.9	61.1	68.2	56.9	53.3	59.7	57.8	47.3
CoDA+EATA	58.4	53.3	53.7	44.5	46.7	47.3	43.7	47.3	50.3	56.7	65.7	53.0	56.9	64.3	62.0	53.6
SAR	1.5	2.0	1.3	41.0	35.1	43.3	35.6	16.8	42.3	54.6	63.4	51.5	38.8	53.4	52.2	35.5
CoDA+SAR	52.8	47.2	47.6	40.2	41.4	42.8	35.4	26.5	46.4	50.0	61.8	47.6	50.7	61.0	58.8	47.3
DeYO	0.2	0.3	0.2	48.1	46.1	51.6	32.0	45.7	52.7	63.4	70.2	58.4	55.4	62.4	59.8	43.1
CoDA+DeYO	20.2	26.0	12.3	47.2	49.3	50.6	29.7	39.2	53.6	59.3	68.5	54.7	59.1	66.5	63.7	46.7
COME	0.2	0.2	0.2	47.6	46.0	51.2	23.1	38.1	52.7	63.6	69.8	58.4	54.1	62.1	59.4	41.8
CoDA+COME	23.2	30.4	12.5	47.0	49.2	50.7	25.7	31.1	53.5	60.5	68.5	56.3	58.3	66.3	63.6	46.4
ZeroSiam	23.6	23.5	25.3	47.6	46.1	51.0	48.1	51.9	53.3	63.4	69.9	58.3	55.9	61.9	59.4	49.3
CoDA+ZeroSiam	59.0	54.3	54.7	47.1	49.6	50.6	47.9	50.5	54.2	59.9	68.2	54.7	59.5	66.2	63.6	56.0
ConvNeXt-T	21.5	27.7	22.2	24.6	10.9	32.5	30.9	44.4	52.5	39.8	72.0	44.7	23.2	17.6	55.7	34.7
EATA	35.7	38.2	36.1	31.4	22.3	37.4	34.0	48.9	49.1	52.3	73.2	53.0	30.1	32.4	56.8	42.1
CoDA+EATA	58.9	54.3	54.2	31.1	31.5	36.7	35.1	46.4	49.7	47.0	69.6	48.7	46.9	57.8	61.0	48.6
SAR	33.4	35.7	33.7	27.3	7.7	34.1	30.6	44.8	47.3	16.6	72.9	49.0	22.1	24.5	56.2	35.7
CoDA+SAR	58.6	53.8	53.5	27.0	23.5	34.2	32.5	43.1	48.5	16.8	69.1	47.2	43.8	56.4	60.4	44.6
DeYO	36.3	38.7	36.7	29.1	3.0	37.0	20.4	50.4	47.9	5.1	73.7	54.9	8.7	23.1	57.3	34.8
CoDA+DeYO	59.1	54.6	54.1	29.3	10.9	36.6	21.9	47.2	49.7	4.0	70.5	48.8	23.7	56.8	61.4	41.9
COME	40.5	42.6	40.3	36.2	2.0	43.3	23.6	54.6	54.0	58.7	75.1	58.4	11.9	43.9	59.9	43.0
CoDA+COME	60.1	55.9	55.8	35.4	9.9	42.1	25.6	51.4	53.6	31.9	72.1	55.1	30.1	61.8	63.8	47.0
ZeroSiam	39.4	41.7	39.5	33.6	27.4	42.0	38.7	53.2	51.7	59.0	73.9	56.9	37.1	44.5	58.3	46.5
CoDA+ZeroSiam	59.7	55.6	55.2	32.8	33.5	40.7	39.3	50.0	52.1	54.4	70.7	51.9	49.9	60.6	61.8	51.2
Swin-T	22.6	23.9	19.4	20.1	12.5	24.7	24.5	38.3	43.9	44.0	65.8	35.8	21.2	7.6	31.3	29.1
EATA	37.0	38.4	37.8	31.3	30.9	39.8	38.5	47.1	47.0	52.7	70.1	49.3	43.2	36.4	48.8	43.2
CoDA+EATA	56.7	53.2	52.8	32.2	36.1	40.1	38.9	45.1	48.4	47.0	67.4	43.7	51.5	57.2	58.3	48.6
SAR	28.4	29.4	29.0	20.8	15.1	27.2	27.6	39.4	37.8	31.0	67.2	44.1	15.0	5.0	40.8	30.5
CoDA+SAR	54.1	50.0	49.8	22.0	25.6	29.1	30.6	39.4	43.1	15.6	64.1	41.7	40.2	45.8	55.0	40.4
DeYO	35.3	36.3	36.3	22.0	24.1	35.8	17.6	42.1	43.0	48.1	70.3	50.1	30.6	33.0	47.9	38.2
CoDA+DeYO	55.8	52.4	52.3	22.1	29.3	37.4	18.2	40.3	45.4	23.8	67.8	43.7	43.6	56.1	58.3	43.1
COME	41.3	43.1	42.4	35.7	9.4	44.9	5.9	53.2	51.8	57.8	72.2	48.0	20.9	43.6	54.4	41.6
CoDA+COME	57.8	54.3	54.0	35.2	26.3	44.8	10.3	50.1	50.8	48.9	69.6	46.1	33.7	57.9	61.3	46.7
ZeroSiam	37.8	40.2	38.5	31.3	33.0	40.5	41.0	48.8	47.0	52.5	69.8	50.6	46.1	42.0	49.9	44.6
CoDA+ZeroSiam	56.6	53.7	53.3	31.6	36.3	41.5	41.8	46.7	49.1	46.6	67.3	44.0	53.1	58.5	58.9	49.3

B.2 Per-Corruption Results under Label Shift With Batch Size 64

Table 10 reports per-corruption results on ImageNet-C at severity 5 under label shift with batch size 64. CoDA improves the average accuracy of the model adaptation baselines across methods and backbones.

Table 10: Per-corruption top-1 accuracy (%) on ImageNet-C at severity 5 under label-shifted test streams with batch size 64.

Model+Method	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
ResNet50-GN	18.0	19.8	17.9	19.8	11.3	21.4	24.9	40.4	47.3	33.6	69.3	36.3	18.6	28.4	52.3	30.6
EATA	33.0	36.3	33.2	22.7	23.7	31.8	35.2	46.2	42.9	51.9	71.2	45.9	35.6	53.5	56.4	41.3
CoDA+EATA	56.2	52.8	52.5	22.7	29.0	31.6	34.6	45.5	45.1	43.9	68.8	38.5	46.5	59.7	59.7	45.8
SAR	28.3	32.9	29.8	20.9	19.7	31.1	29.1	41.1	43.7	45.6	70.8	44.0	16.5	48.9	55.1	37.2
CoDA+SAR	56.9	52.8	52.4	20.4	26.5	30.6	31.1	41.8	46.3	35.9	67.3	34.6	38.7	57.8	58.8	43.5
DeYO	13.4	41.7	38.9	23.8	25.0	39.3	7.8	51.2	49.8	57.1	72.9	50.4	44.8	56.1	57.7	42.0
CoDA+DeYO	47.3	55.4	54.8	22.8	29.0	38.4	11.5	48.9	50.2	43.2	70.4	40.5	52.8	61.6	61.4	45.9
COME	20.4	21.9	20.3	19.7	12.9	21.9	25.2	41.0	47.5	33.7	69.0	35.3	20.9	33.4	52.7	31.7
CoDA+COME	56.1	51.2	50.9	19.5	24.0	22.5	26.4	39.4	44.6	18.1	63.6	27.5	42.3	52.9	57.1	39.7
ZeroSiam	41.7	43.8	42.8	34.1	33.2	43.0	45.7	56.7	55.0	62.0	73.4	54.0	51.8	61.5	61.0	50.7
CoDA+ZeroSiam	53.3	53.5	52.8	33.5	37.8	42.1	44.9	55.2	56.0	58.3	70.8	49.9	57.1	64.9	64.1	52.9
ViT-B	9.5	6.8	8.2	29.0	23.4	33.9	27.1	15.9	26.5	47.2	54.6	44.1	30.5	44.5	47.8	29.9
EATA	45.4	45.6	45.4	53.0	53.0	56.8	53.9	61.1	59.2	69.1	75.5	57.5	61.3	69.0	65.6	58.1
CoDA+EATA	63.8	59.3	59.7	51.4	55.1	56.1	53.2	57.2	57.7	63.8	73.1	58.7	62.2	70.3	67.2	60.6
SAR	43.5	43.0	44.8	53.1	50.3	56.1	51.4	59.0	57.7	66.6	74.8	64.4	55.9	66.9	64.0	56.8
CoDA+SAR	63.5	58.3	59.2	52.4	53.4	55.1	50.2	55.2	57.6	62.5	72.8	60.3	59.8	69.0	65.9	59.7
DeYO	52.2	52.5	53.3	57.6	58.3	62.8	59.5	67.0	65.5	73.2	77.9	67.5	68.2	73.0	69.3	63.9
CoDA+DeYO	65.5	61.7	62.1	56.9	58.7	61.5	58.1	63.9	63.5	70.3	76.1	64.2	67.8	73.3	70.2	64.9
COME	49.0	49.9	50.8	55.4	56.2	60.8	58.0	64.9	63.0	71.6	76.7	66.5	66.4	71.6	67.9	61.9
CoDA+COME	64.1	60.0	60.3	54.4	57.0	60.0	56.7	63.2	62.0	69.3	75.7	63.6	67.0	72.0	69.1	63.6
ZeroSiam	51.3	51.5	52.2	57.0	57.1	61.8	59.5	65.4	64.9	72.4	77.3	67.7	66.3	71.6	68.5	63.0
CoDA+ZeroSiam	65.5	61.7	62.2	56.1	58.1	60.6	58.2	61.9	62.9	67.5	75.5	63.2	66.5	72.4	69.6	64.1
ViT-S	2.0	2.1	1.5	24.1	17.2	30.3	22.1	9.5	19.3	37.8	44.4	30.0	24.8	38.2	41.1	23.0
EATA	16.9	16.1	19.1	43.5	41.8	45.4	41.3	42.7	45.9	59.0	65.1	54.8	52.0	57.7	56.5	43.9
CoDA+EATA	57.5	52.0	52.5	42.6	45.2	44.8	40.6	45.0	48.4	53.6	63.7	50.2	56.0	63.3	61.6	51.8
SAR	1.6	2.4	1.4	41.6	34.8	43.7	35.8	13.6	43.0	54.9	63.6	51.7	39.8	53.4	52.6	35.6
CoDA+SAR	52.2	48.0	47.3	40.7	41.6	43.4	36.1	21.4	46.9	49.9	61.9	46.9	51.0	61.0	59.0	47.2
DeYO	0.2	0.3	0.2	47.8	46.6	51.4	7.2	50.0	53.2	63.2	69.8	58.1	56.6	62.1	59.6	41.8
CoDA+DeYO	13.0	16.3	4.4	47.0	49.5	50.5	8.5	46.9	53.3	59.1	68.2	54.1	59.6	66.3	63.7	44.0
COME	0.2	0.2	0.2	46.8	46.1	50.6	7.4	40.4	52.6	62.7	69.1	57.2	55.3	61.4	58.7	40.6
CoDA+COME	11.8	10.5	4.9	45.9	48.5	49.9	8.7	33.1	53.1	58.9	67.8	53.8	58.7	65.6	63.0	42.3
ZeroSiam	21.0	20.7	22.8	46.7	45.2	50.1	45.7	49.9	52.3	62.1	68.6	57.4	54.6	60.7	58.4	47.7
CoDA+ZeroSiam	59.4	54.5	54.6	47.0	50.1	50.7	47.9	50.8	54.8	60.9	68.8	54.8	60.6	66.7	64.1	56.4
ConvNeXt-T	21.5	27.7	22.2	24.6	10.9	32.5	30.9	44.4	52.5	39.8	72.0	44.7	23.2	17.6	55.7	34.7
EATA	34.5	36.6	35.4	30.8	19.7	36.6	32.8	47.9	48.3	52.0	73.1	51.1	28.0	29.8	56.7	40.9
CoDA+EATA	58.3	53.8	53.5	30.2	30.4	35.8	33.9	45.2	49.4	44.2	69.3	46.8	45.0	57.3	60.9	47.6
SAR	33.3	35.6	33.7	27.8	10.3	33.8	30.6	44.5	47.1	46.4	72.9	48.9	22.0	23.9	56.2	37.8
CoDA+SAR	58.1	53.5	53.2	27.5	26.2	33.9	32.5	42.7	48.5	39.6	68.8	46.6	42.8	56.3	60.3	46.0
DeYO	36.1	38.6	36.4	33.1	14.8	37.3	13.7	50.2	48.9	56.0	73.7	54.6	16.0	30.6	57.4	39.8
CoDA+DeYO	58.8	54.5	54.1	32.1	27.9	36.5	16.8	46.9	49.8	30.8	70.4	49.5	29.2	57.5	61.3	45.1
COME	40.1	42.2	39.9	36.5	24.8	43.8	24.9	54.9	53.8	60.7	75.0	58.2	23.4	45.8	59.9	45.6
CoDA+COME	60.0	55.8	55.3	35.3	36.6	42.6	27.1	51.5	53.5	50.2	72.2	55.1	45.0	62.1	63.7	51.1
ZeroSiam	40.2	42.1	39.9	35.5	27.8	42.6	39.1	53.6	51.9	59.9	74.1	57.3	36.8	45.6	58.7	47.0
CoDA+ZeroSiam	59.5	55.4	54.9	34.6	36.0	41.4	39.3	50.0	52.2	52.5	71.0	50.7	49.6	60.9	62.2	51.3
Swin-T	22.6	23.9	19.4	20.1	12.5	24.7	24.5	38.3	43.9	44.0	65.8	35.8	21.2	7.6	31.3	29.1
EATA	32.5	34.9	33.0	29.5	27.6	37.2	35.3	44.7	44.4	50.1	69.9	47.3	40.0	31.6	46.1	40.3
CoDA+EATA	55.6	51.8	51.6	30.7	32.7	37.5	36.0	43.3	46.5	43.8	67.0	40.5	49.5	55.6	57.5	46.7
SAR	28.6	28.5	28.4	21.4	15.9	28.4	25.8	39.2	37.8	32.3	67.3	43.8	15.8	4.9	41.0	30.6
CoDA+SAR	53.9	49.8	49.9	23.3	26.3	30.1	30.1	39.5	42.9	23.9	64.3	41.6	39.4	47.0	55.1	41.1
DeYO	35.0	37.0	35.7	25.9	24.5	39.0	12.8	45.4	43.4	49.9	70.1	49.6	41.9	5.8	48.5	37.6
CoDA+DeYO	55.9	52.5	52.0	27.0	29.4	39.5	13.4	42.7	46.5	34.2	67.6	43.6	50.6	35.2	58.4	43.2
COME	40.3	42.3	41.6	35.1	2.8	44.9	3.7	52.9	51.2	58.5	71.9	49.4	50.4	5.3	53.9	40.3
CoDA+COME	57.5	53.9	53.7	35.3	7.5	45.0	5.0	49.9	50.9	55.2	69.5	43.8	55.3	37.1	61.2	45.4
ZeroSiam	39.2	40.5	39.1	31.8	33.9	41.7	41.8	48.9	47.8	53.7	70.3	50.7	47.1	41.8	50.1	45.2
CoDA+ZeroSiam	57.0	54.2	53.5	32.9	37.9	42.3	42.2	46.9	49.5	45.8	68.0	43.0	53.8	58.6	59.1	49.7

B.3 Per-Corruption Results under Label Shift with Batch Size One

Table 11 reports per-corruption results on ImageNet-C at severity 5 under label shift with batch size one. CoDA improves the average accuracy of the model adaptation baselines across methods and backbones. These results show that the improvements remain when each update is based on a single test sample.

Table 11: Per-corruption top-1 accuracy (%) on ImageNet-C at severity 5 under label-shifted test streams with batch size 1.

Model+Method	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
ResNet50-GN	18.0	19.8	17.9	19.8	11.3	21.4	24.9	40.4	47.3	33.6	69.3	36.3	18.6	28.4	52.3	30.6
EATA	19.4	21.7	19.4	18.5	13.5	23.8	25.5	39.4	41.7	35.4	69.2	38.6	20.7	35.7	53.2	31.7
CoDA+EATA	55.6	50.8	50.6	18.8	24.0	24.0	27.2	39.4	44.1	22.8	64.8	33.9	41.1	53.3	57.4	40.5
SAR	22.9	26.4	23.9	19.5	15.7	29.7	30.5	47.0	45.1	28.4	72.3	44.7	15.0	47.5	56.1	35.0
CoDA+SAR	56.3	51.9	51.6	19.2	25.5	29.2	31.6	45.5	46.7	17.0	69.2	40.3	35.4	58.6	59.8	42.5
DeYO	41.0	45.8	43.4	25.7	22.3	43.3	7.5	55.3	53.2	60.0	73.3	54.1	52.2	61.3	60.4	46.6
CoDA+DeYO	59.4	56.5	56.1	24.4	27.4	42.9	13.5	53.3	53.5	57.6	71.3	50.1	56.8	64.1	63.2	50.0
COME	17.8	19.7	17.7	19.7	11.3	21.3	24.9	40.3	47.8	33.6	68.8	35.6	18.5	27.8	51.7	30.4
CoDA+COME	56.0	51.0	50.8	19.5	23.9	22.4	26.3	39.2	44.3	18.2	62.8	28.0	41.9	52.5	57.0	39.6
ZeroSiam	38.8	39.9	39.2	33.0	33.3	41.3	46.1	54.7	52.4	61.1	73.0	52.8	51.1	59.9	59.4	49.1
CoDA+ZeroSiam	58.3	55.1	55.3	33.5	37.7	42.3	44.4	51.4	52.7	51.2	70.3	44.9	56.5	63.8	63.1	52.0
ViT-B	9.5	6.8	8.2	29.0	23.4	33.9	27.1	15.9	26.5	47.2	54.6	44.1	30.5	44.5	47.8	29.9
EATA	17.8	13.8	17.3	35.6	28.7	38.8	32.5	22.6	31.6	52.2	60.8	51.7	35.0	51.4	51.1	36.1
CoDA+EATA	60.1	52.1	53.9	34.5	36.7	38.0	31.0	25.7	37.1	48.5	57.1	47.2	48.0	59.6	58.1	45.9
SAR	24.6	25.9	24.7	43.6	36.1	44.6	36.4	35.8	42.5	53.7	62.8	54.4	39.0	55.1	54.2	42.2
CoDA+SAR	60.7	54.2	55.1	44.5	45.7	49.1	41.6	42.8	49.1	53.7	64.6	53.1	52.7	62.6	60.7	52.7
DeYO	53.8	54.4	55.2	59.0	60.0	64.6	5.7	68.6	66.7	73.8	78.3	68.6	69.9	74.1	70.7	61.6
CoDA+DeYO	66.0	62.7	61.9	58.6	60.3	63.7	6.2	65.6	64.8	71.7	76.6	65.8	69.0	74.1	71.3	62.5
COME	50.7	51.4	52.2	56.9	57.3	62.6	7.6	66.1	63.9	72.4	77.0	67.7	68.0	72.5	69.1	59.7
CoDA+COME	64.8	60.6	61.1	56.2	57.7	61.6	7.9	64.2	62.7	70.2	76.1	64.4	68.2	72.8	69.8	61.2
ZeroSiam	49.4	49.2	50.5	56.1	55.5	60.3	57.9	64.2	63.6	71.2	76.9	67.3	63.8	70.5	67.4	61.6
CoDA+ZeroSiam	65.5	61.5	62.2	55.6	58.2	60.2	58.1	61.9	62.8	68.1	75.4	63.3	66.3	72.3	69.5	64.1
ViT-S	2.0	2.1	1.5	24.1	17.2	30.3	22.1	9.5	19.3	37.8	44.4	30.0	24.8	38.2	41.1	23.0
EATA	2.2	2.3	1.7	26.6	19.9	31.9	23.6	11.2	21.3	40.8	46.2	36.6	26.7	41.4	43.2	25.0
CoDA+EATA	57.0	50.7	51.3	26.1	28.6	31.6	23.8	18.3	30.3	35.4	45.5	32.6	43.8	53.6	53.8	38.8
SAR	2.1	2.2	1.6	35.2	26.6	39.7	29.8	22.6	30.8	43.5	55.3	37.9	34.0	47.0	45.3	30.2
CoDA+SAR	56.9	50.7	51.2	39.4	40.1	42.6	32.9	27.9	40.3	42.3	56.3	40.0	46.9	57.5	56.0	45.4
DeYO	0.5	0.8	0.6	49.7	49.3	53.7	5.9	53.7	55.4	65.0	70.8	59.9	59.1	64.1	61.0	43.3
CoDA+DeYO	43.5	41.4	39.7	48.8	51.4	52.6	6.8	50.6	54.9	62.3	69.2	56.8	61.0	67.4	64.4	51.4
COME	0.5	3.2	0.7	47.8	48.1	52.3	6.0	51.1	54.2	63.9	70.1	58.7	57.9	63.3	60.0	42.5
CoDA+COME	46.6	51.5	42.0	47.1	49.9	51.7	6.8	47.6	53.8	58.1	68.8	55.8	60.0	66.4	63.8	51.3
ZeroSiam	21.5	21.5	23.9	45.5	43.3	48.6	44.9	48.9	51.4	61.2	67.8	56.3	52.6	59.4	57.1	46.9
CoDA+ZeroSiam	58.8	53.6	54.2	44.8	48.2	49.0	46.4	49.2	53.6	57.2	67.1	52.3	59.1	65.4	62.8	54.8
ConvNeXt-T	21.5	27.7	22.2	24.6	10.9	32.5	30.9	44.4	52.5	39.8	72.0	44.7	23.2	17.6	55.7	34.7
EATA	25.6	29.7	26.2	24.8	11.1	32.6	30.8	44.5	50.6	40.0	72.2	45.8	23.4	17.9	55.7	35.4
CoDA+EATA	57.4	52.6	52.1	24.4	25.9	32.6	31.9	42.4	49.8	27.1	67.7	37.5	43.3	55.3	60.2	44.0
SAR	29.7	33.7	29.6	26.1	11.2	33.4	30.8	44.8	47.5	38.5	73.0	46.0	22.7	18.2	56.2	36.1
CoDA+SAR	58.0	53.1	52.6	25.6	26.3	33.5	32.3	43.1	48.5	26.3	69.1	39.7	43.1	55.7	60.3	44.5
DeYO	35.0	39.2	35.8	23.2	9.4	38.6	10.3	51.9	49.4	57.5	73.9	55.6	15.0	26.8	58.0	38.6
CoDA+DeYO	58.4	54.8	54.0	22.6	22.4	37.6	14.5	48.9	50.8	54.8	70.8	54.5	30.0	57.0	61.7	46.2
COME	41.0	43.3	41.1	37.8	9.1	45.7	13.5	57.1	55.9	62.6	75.2	59.4	26.7	48.1	61.2	45.2
CoDA+COME	60.1	56.1	55.7	36.6	24.5	44.3	20.5	53.7	55.1	60.1	72.5	58.1	46.1	62.5	64.3	51.3
ZeroSiam	41.0	43.3	40.8	36.4	32.8	43.4	41.1	54.2	53.1	60.6	74.3	58.4	40.0	49.5	59.2	48.5
CoDA+ZeroSiam	59.5	55.8	55.3	35.5	38.5	42.1	40.6	50.6	53.0	54.3	71.4	53.4	50.8	61.6	62.6	52.3
Swin-T	22.6	23.9	19.4	20.1	12.5	24.7	24.5	38.3	43.9	44.0	65.8	35.8	21.2	7.6	31.3	29.1
EATA	24.0	25.0	20.8	20.6	13.2	25.4	24.7	37.9	41.4	43.5	66.3	37.4	21.9	8.0	32.3	29.5
CoDA+EATA	52.9	48.4	48.2	21.4	24.1	26.9	26.6	37.8	43.2	23.3	62.1	31.8	42.0	49.3	51.5	39.3
SAR	25.1	27.2	22.9	21.3	14.3	27.7	27.7	38.9	37.7	44.1	67.3	38.0	21.0	8.2	39.1	30.7
CoDA+SAR	53.8	49.6	49.1	23.0	25.9	28.8	30.0	38.9	42.4	22.9	64.1	32.1	42.1	51.4	54.7	40.6
DeYO	34.1	37.0	34.6	27.1	24.6	39.6	12.0	47.4	42.8	50.8	70.4	50.6	42.4	20.5	49.4	38.9
CoDA+DeYO	55.2	52.6	52.1	27.6	28.9	40.5	11.2	44.2	47.0	47.8	68.0	49.9	51.1	53.3	58.7	45.9
COME	39.6	41.7	40.4	35.3	6.6	45.3	2.9	53.7	51.6	57.9	72.1	51.3	50.6	37.8	53.5	42.7
CoDA+COME	57.5	53.8	53.5	35.0	34.2	45.4	5.9	50.2	51.1	50.6	69.7	42.3	55.5	59.2	61.0	48.3
ZeroSiam	38.6	39.7	38.2	30.0	34.5	40.9	42.6	49.5	47.4	53.9	70.6	50.2	48.4	42.5	50.0	45.1
CoDA+ZeroSiam	56.9	54.1	53.6	32.1	38.5	42.2	42.8	47.4	50.1	47.9	68.5	43.8	54.5	58.5	59.4	50.0

B.4 CIFAR-10-C Results

Table 12 reports results on CIFAR-10-C at severity 5 using WRN-28-10 with I-DDPM restoration. CoDA improves TENT, SAR, and DeYO by 0.4, 0.5, and 0.5 percentage points.

Table 12: Top-1 accuracy (%) on CIFAR-10-C at severity 5.

Method	Original	+Ours
Source	56.7	–
TENT	82.3	**82.7**
SAR	81.5	**82.0**
DeYO	82.3	**82.8**

C More Experimental Results

C.1 Alternative Restoration Models

Experimental setup. In the main experiments, we use a Diffusion model as the default restoration model for constructing the prior view. To test whether CoDA depends on this choice, we replace Diffusion model with Restormer (Zamir et al., 2022) and AdaIR (Cui et al., 2025). For AdaIR, we use the all-in-one checkpoint trained for multiple degradation types, while for Restormer we use degradation-specific checkpoints trained for individual degradation types. We therefore evaluate three ImageNet-C corruptions, defocus blur, motion blur, and snow, for which the corresponding or closely related Restormer restoration models are available. These models are used only to produce the prior view, while the observation branch still receives the corrupted input. All other parts of CoDA remain unchanged, including Cross-View Gating, Selective Coupling with Jensen–Shannon Divergence, and Entropy-Aware Prediction Fusion. We evaluate at severity 5 using ViT-S and ConvNeXt-Tiny, and report average top-1 accuracy. Figure 6 shows examples of the selected corruption types and the restored images produced by different restoration models.

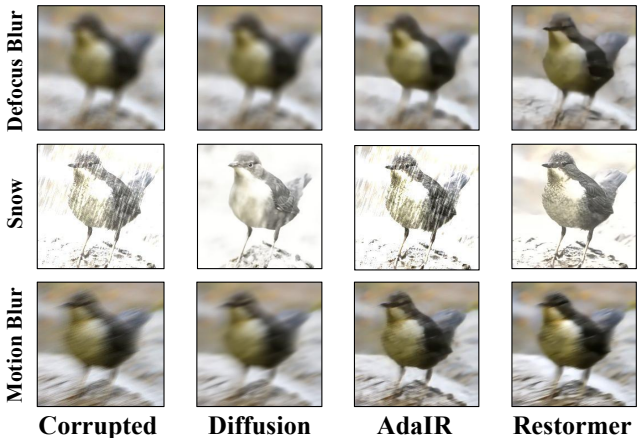


Figure 6: Examples of restored outputs from different restoration models. From top to bottom, the degradation types are defocus blur, snow, and motion blur.

Results. Table 13 compares four settings for each restoration model under the same evaluation protocol. *Source* denotes the source classifier evaluated directly on corrupted images without test-time adaptation. *Data adaptation* follows the DDA pipeline, but replaces the original diffusion restoration model with Restormer or AdaIR, while keeping the classifier fixed for prediction. *ZeroSiam* denotes model-based TTA on corrupted inputs. *CoDA+ZeroSiam* uses the same restored images as the prior view and jointly adapts them with the corrupted observations. This comparison separates data adaptation, model adaptation, and their combination in CoDA.

Table 13: Results with alternative restoration models on defocus blur, motion blur, and snow at severity 5. Data adaptation follows the DDA pipeline, with the diffusion restoration model replaced by Restormer or AdaIR. The same restored images are used as the prior view in CoDA+ZeroSiam.

Restormer

Method	ViT-S	ConvT
Source	21.28	33.84
Data adaptation	41.03	52.44
ZeroSiam	50.28	43.84
CoDA+ZeroSiam	61.67	57.83

AdaIR

Method	ViT-S	ConvT
Source	21.28	33.84
Data adaptation	24.91	38.20
ZeroSiam	50.28	43.84
CoDA+ZeroSiam	52.47	46.75

Analysis. With Restormer, CoDA+ZeroSiam reaches 61.67% on ViT-S and 57.83% on ConvNeXt-Tiny, improving over ZeroSiam by 11.39 and 13.99 points. With AdaIR, the corresponding gains are 2.19 and 2.91 points. The larger gains with Restormer suggest that the quality of the prior view still

affects the final improvement, while the gains with both models show that CoDA is not limited to diffusion-based restoration.

Using non-diffusion restoration also reduces the restoration cost. In our setting, AdaIR takes about 0.8 s at batch size 8, which is substantially faster than the diffusion-based restoration used in the main experiments. This shows that CoDA can work with faster restoration models while keeping the same co-adaptation framework.

C.2 Cross-View Gating

Table 14 provides the detailed results for Cross-View Gating reported in Figure 4. We compare SAR-based CoDA with and without Cross-View Gating across five backbones, while keeping the other components unchanged. We report both top-1 accuracy and the update ratio, defined as the percentage of samples used for model updates.

Table 14: Impact of Cross-View Gating on SAR-based CoDA. Ratio denotes the percentage of samples selected for model updates.

Metric	Conv-T	ResNet50	Swin-T	ViT-B	ViT-S
w/o CVG Acc.	45.01	39.60	39.02	56.94	46.07
w/o CVG Ratio (%)	46.10	45.17	56.16	70.45	59.42
w/ CVG Acc.	45.11	42.00	39.34	59.16	46.07
w/ CVG Ratio (%)	32.45	29.14	33.82	57.70	36.98

Cross-View Gating reduces the update ratio on all five backbones while maintaining or improving accuracy. For example, on ResNet50, the update ratio decreases from 45.17% to 29.14%, while accuracy increases from 39.60% to 42.00%. These results further show that Cross-View Gating achieves comparable or better adaptation performance with fewer model updates.

C.3 Additional Dual-Path Ablations at Batch Size One

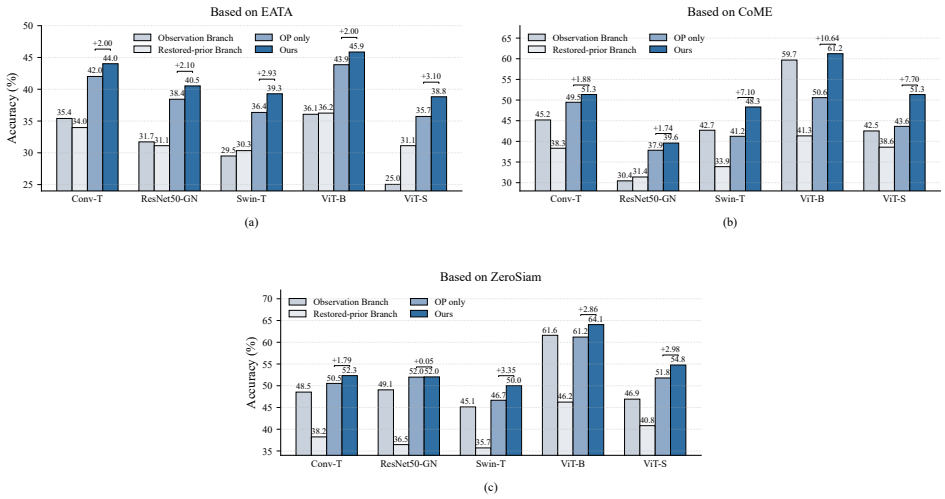


Figure 7: Batch-size-one dual-path ablations on ImageNet-C at severity 5. The main paper reports SAR and DeYO; here we provide the remaining results under the same setting. We compare the observation branch, the restored-prior branch, a direct observation-prior variant, and full CoDA. Full CoDA gives more stable performance across base TTA methods and backbones.

The main paper reports batch-size-one dual-path results for SAR and DeYO. Here, we provide the corresponding results for EATA, COME, and ZeroSiam on ImageNet-C at severity 5 under the same setting. We compare four variants: the observation branch, the restored-prior branch, their direct combination (OP only), and full CoDA.

Analysis. Figure 7 shows a similar pattern across the additional base TTA methods. The observation and restored-prior branches have different strengths, while either branch alone can be unreliable under batch size one. Directly combining the two views (OP only) can improve over a single branch in some cases, but the gains are not always consistent. In contrast, full CoDA consistently improves over OP only across the evaluated methods and backbones.

By filtering unsupported updates and coupling the two branches only on agreed samples, CoDA better controls the interaction between the two views. These results further show that the restored view provides useful prior information, but directly combining it with the corrupted observation is not always sufficient.

D Additional Analyses and Evaluations

D.1 Sensitivity to Top-k and Fusion Strategies

We evaluate $k \in \{1, 5, 20, 50, 100\}$ on ImageNet-C at severity 5 using ZeroSiam with ResNet50. We compare uniform logit fusion, minimum-entropy branch selection, entropy soft fusion using $1 - H(p)$, and our Entropy-Aware Prediction Fusion.

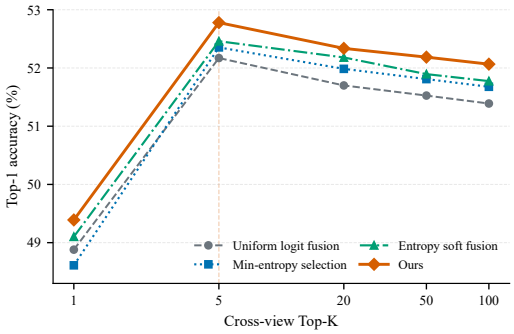


Figure 8: Top-k sensitivity and fusion-strategy comparison on ImageNet-C using ZeroSiam with ResNet50.

Analysis. As shown in Figure 8, $k = 5$ gives the highest accuracy across all four fusion strategies, supporting our default choice. Entropy-Aware Prediction Fusion also consistently improves over uniform logit fusion and entropy-based soft fusion across the tested values of k, with gains of 0.51–0.68 and 0.16–0.32 percentage points, respectively. These results suggest that entropy-aware prediction fusion can provide an additional benefit in the evaluated configuration.

D.2 Extension to Vision-Language Models

Table 15: Per-corruption top-1 accuracy (%) on ImageNet-C at severity 5 using CLIP ViT-B/32. We compare the source model, DDA for data adaptation, CLIPArTT for model adaptation, their independent combination DDA+CLIPArTT, and CoDA+CLIPArTT.

Model+Method	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
Source	11.09	11.43	11.16	21.47	11.27	19.32	17.24	22.09	23.04	26.08	45.53	16.41	16.81	28.15	25.86	20.47
DDA	36.40	33.97	33.50	21.01	20.07	19.00	17.74	22.89	27.47	19.05	42.34	11.04	30.75	38.56	38.81	27.51
CLIPArTT	19.01	20.35	19.30	**24.48**	20.28	**26.45**	23.35	**27.17**	26.12	**34.39**	**46.24**	**22.58**	23.27	32.92	31.34	26.48
DDA+CLIPArTT	37.94	36.08	35.56	24.19	24.74	25.93	23.22	26.31	29.53	25.58	43.74	15.85	33.29	40.88	40.56	30.89
CoDA+CLIPArTT	**39.30**	**36.78**	**36.31**	24.27	**26.10**	26.18	**23.55**	26.95	**30.43**	31.84	44.86	20.06	**34.05**	**41.18**	**41.29**	**32.21**

Experimental setup. We evaluate CoDA on ImageNet-C at severity 5 using CLIP ViT-B/32 and CLIPArTT (A Vargas Hakim et al., 2025) as the base model-adaptation method. Table 15 compares the source model, DDA for data adaptation, CLIPArTT for model adaptation, their independent combination DDA+CLIPArTT, and CoDA+CLIPArTT. We report top-1 accuracy for each corruption type and the average over all 15 types.

Analysis. Table 15 shows that CoDA+CLIPArTT achieves 32.21% average accuracy, exceeding CLIPArTT and DDA+CLIPArTT by 5.73 and 1.32 percentage points. Improvements over the independent combination occur on all 15 corruption types, supporting the applicability of controlled co-adaptation to CLIP-based classification.

D.3 Preservation of Clean-Domain Performance

Clean-domain stability. We first evaluate CoDA+ZeroSiam directly on the clean ImageNet validation set to examine whether applying the adaptation pipeline in the absence of distribution shift introduces undesirable model drift.

Table 16: Top-1 accuracy (%) on the clean ImageNet validation set without distribution shift. CoDA is instantiated with ZeroSiam across five backbone architectures.

Method	ResNet-50	ViT-B	ViT-S	Conv-T	Swin-T
Source	79.97	78.01	74.64	82.10	80.91
CoDA+ZeroSiam	78.77	82.39	78.25	81.39	79.75

As shown in Table 16, CoDA largely maintains the source-model performance when applied directly to clean inputs. It improves accuracy by 4.38 and 3.61 percentage points on ViT-B and ViT-S, while the remaining three backbones exhibit modest decreases of 0.71–1.20 points. These results suggest that CoDA does not introduce substantial performance degradation when no distribution shift is present.

Source-domain preservation after adaptation. We further examine whether adaptation to corrupted target domains compromises the model's original source-domain performance. For each ImageNet-C corruption at severity 5, we first adapt CoDA+ZeroSiam on the corresponding corrupted domain and then evaluate the adapted model on the clean ImageNet validation set. This setting directly measures how well the original clean-domain capability is preserved after corruption-specific adaptation.

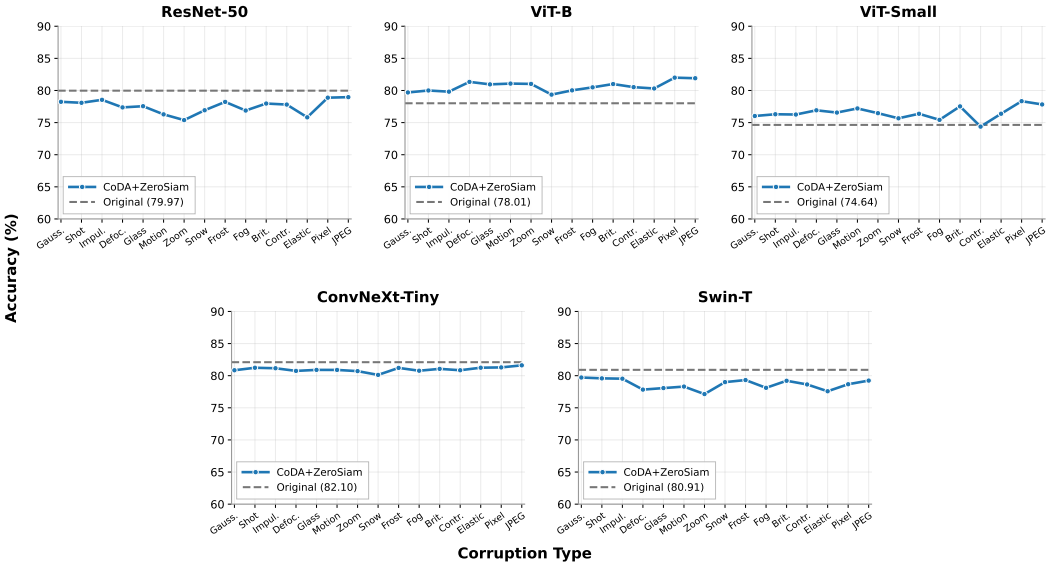


Figure 9: **Clean ImageNet performance after corruption-specific adaptation.** CoDA+ZeroSiam is adapted separately to each of the 15 ImageNet-C corruption types at severity 5 and subsequently evaluated on the clean ImageNet validation set. Results are reported across five backbone architectures. The dashed line denotes the clean accuracy of the original source model before test-time adaptation.

As shown in Figure 9, CoDA largely preserves clean-domain performance after corruption-specific adaptation across corruption types and backbone architectures. The adapted models remain close

to the original source-model accuracy, showing that the gains under corruption shifts are achieved without substantial degradation on the clean source domain.

E LIMITATIONS

CoDA still has two practical limitations. First, its runtime is primarily constrained by the restoration step, since each test sample needs to be paired with a restored view before online adaptation. Although the interaction overhead introduced by CoDA itself is small and faster restoration backends can substantially reduce the overall cost, restoration remains the main computational bottleneck. Second, the effectiveness of co-adaptation depends on the quality of the restored view. If restoration leaves substantial corruption or introduces semantic distortion, the prior-informed branch becomes less reliable and the benefit of co-adaptation can be reduced. Our experiments with alternative restoration models such as Restormer and AdaIR show that CoDA is not tied to a specific restoration architecture, but also suggest that its gains vary with the quality of the restored prior. Improving the efficiency and reliability of restoration therefore remains an important direction for further improving CoDA.