Panda: Test-Time Adaptation with Negative Data Augmentation

Ruxi Deng[1], **Wenxuan Bao**[1], **Tianxin Wei**[1], **Jingrui He**[1]

[1]University of Illinois Urbana-Champaign
{ruxid2, wbao4, twei10, jingrui}@illinois.edu

Abstract

Pretrained vision-language models exhibit strong zero-shot classification capabilities, but their predictions degrade significantly under common image corruptions. To improve robustness, many test-time adaptation (TTA) methods adopt positive data augmentation (PDA), which generates multiple views of each test sample to reduce prediction variance. However, these methods suffer from two key limitations. First, it introduces considerable computational overhead due to the large number of augmentations required per image. Second, it fails to mitigate prediction bias, where the model tends to predict certain classes disproportionately under corruption, as PDA operates on corrupted inputs and typically does not remove the corruption itself. To address these challenges, we propose Panda, a novel TTA method based on negative data augmentation (NDA). Unlike positive augmentations that preserve object semantics, Panda generates negative augmentations by disrupting semantic content. It divides images into patches and randomly assembles them from a shared patch pool. These negatively augmented images retain corruption-specific features while discarding object-relevant signals. We then subtract the mean feature of these negative samples from the original image feature, effectively suppressing corruption-related components while preserving class-relevant information. This mitigates prediction bias under distribution shifts. Importantly, Panda allows augmentation to be shared across samples within a batch, resulting in minimal computational overhead. Panda can be seamlessly integrated into existing test-time adaptation frameworks and substantially improve their robustness. Our experiments indicate that Panda delivers superior performance compared to PDA methods, and a wide range of TTA methods exhibit significantly enhanced performance when integrated with Panda.

Code — https://github.com/ruxideng/Panda

1 Introduction

Pretrained vision-language models (VLMs) such as CLIP (Radford et al. 2021) have demonstrated strong zero-shot generalization across a wide range of vision tasks (Zhong et al. 2022; Rao et al. 2022; Patashnik et al. 2021). However, their performance can degrade significantly under image corruptions (Hendrycks and Dietterich 2019). One major reason is that distribution shifts caused by corruptions introduce prediction bias: the model tends to misclassify corrupted images into specific categories (Niu et al. 2023; Lee et al. 2024). This bias arises because corruption patterns may be inadvertently used as spurious features, leading the model to associate them with particular classes regardless of the underlying object content (Hendrycks and Dietterich 2019; Lee et al. 2024).

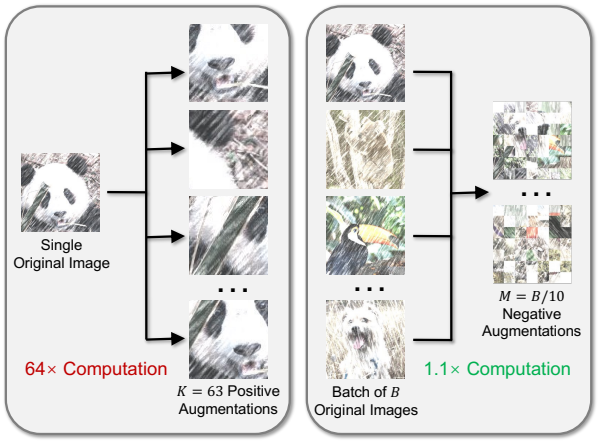

Figure 1: Comparison of positive (left) and negative (right) data augmentation. Left: PDA used in previous TTA algorithms generates K class-preserving views per image, resulting in high computational cost. Right: NDA used in Panda generates M class-agnostic corrupted views shared across a batch of B images, incurring minimal overhead.

To mitigate the impact of image corruptions, recent work has explored test-time adaptation (TTA), which adapts the model on-the-fly using only unlabeled test data. Among various strategies, one popular line of methods is based on positive data augmentation (PDA) (Hendrycks et al. 2020). These methods generate multiple random augmented views for each test image, aggregate the logits to improve prediction robustness (Döbler et al. 2024; Farina et al. 2024) or update the model (Zhang, Levine, and Finn 2022; Shu et al. 2022; Sui, Wang, and Yeung-Levy 2025). However, PDA-based TTA suffers from several limitations. In terms of computational efficiency, it requires generating K aug-

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

mented views independently for each test sample (typically $K = 63$), leading to significant increases in test-time cost. In terms of effectiveness, while PDA preserves semantic content, it also tends to retain corruption patterns. Even after averaging multiple augmented views, these corruptions typically persist and can often amplify prediction bias (see our results in Figure 2). Consequently, existing PDA-based TTA methods are generally ineffective in mitigating prediction bias.

Motivated by these limitations, we propose `Panda`: test-time adaPtAtion with Negative Data Augmentation. Unlike PDA, which preserves object semantics, our negative data augmentation (NDA) intentionally *disrupts semantic content while retaining corruption characteristics*. An illustration is provided in Figure 1. Specifically, `Panda` divides all images within a test batch into small patches to create a shared patch pool, then randomly samples patches from this pool to assemble new, recombined negative augmentations. These negatively augmented images obscure object semantics but preserve the underlying corruption patterns. By subtracting the features of these negatively augmented images from the original image embeddings, `Panda` effectively suppresses corruption-related signals while retaining class-relevant information, thus mitigating prediction bias (see Figure 2).

Furthermore, since negative augmentations are shared across samples within the same batch, `Panda` avoids the computational overhead of generating augmentations for each individual image, making it more efficient than PDA-based TTA methods. Additionally, as `Panda` modifies only the forward propagation step, it can be seamlessly integrated with existing TTA frameworks. Our evaluation on standard corruption benchmarks shows that `Panda` consistently enhances performance across various TTA algorithms with minimal computational cost. We summarize our contributions as follows:

- We identify that existing TTA methods based on positive data augmentation often incur high inference costs while failing to effectively reduce prediction bias.
- We propose `Panda`, a novel TTA method that leverages negative data augmentation to suppress corruption-related features in image embeddings and significantly reduce prediction bias.
- Extensive experiments show that `Panda` achieves better performance than existing PDA-based methods with substantially lower computational cost. In addition, `Panda` can be integrated with most existing TTA algorithms to significantly enhance their robustness.

2 Related Works

Data augmentation refers to the process of generating additional samples by applying transformations to existing data, and is widely used during model training to improve generalization. Broadly, data augmentation methods can be categorized into positive and negative augmentation. Positive data augmentation (PDA) operates on a single image and aims to preserve its semantic content. Common PDA techniques such as Cutout (Devries and Taylor 2017) and AugMix (Hendrycks et al. 2020) act as regularizers that encourage robustness to perturbations. In contrast, negative data augmentation (NDA) typically involves combining information from multiple images and often alters their semantics. Examples include MixUp (Zhang et al. 2018) and CutMix (Yun et al. 2019), which have been shown to further enhance generalization by promoting smoother decision boundaries. It is important to note that these methods are primarily designed for use during model training, rather than at test time.

Test-time adaptation (TTA) adapts a pre-trained model to an unlabeled target domain without accessing source data (Liang, He, and Tan 2024; Xiao and Snoek 2024; Bao et al. 2025b). A prominent line of work, exemplified by Tent (Wang et al. 2021), performs entropy minimization by updating the model's normalization layers. Follow-up methods such as ETA/EATA (Niu et al. 2022), SAR (Niu et al. 2023), and DeYO (Lee et al. 2024) improve adaptation stability by incorporating entropy-aware sample selection or weighting strategies. Notably, DeYO also leverages negative data augmentation to guide this process, using the prediction difference between the original image and its negatively augmented counterpart to assess reliability. Training-free TTA methods (Zhang et al. 2024; Karmanov et al. 2024; Bao et al. 2025a) avoids any model updates and instead modifies predictions directly based on inter-sample similarity. Several TTA approaches incorporate positive data augmentation (PDA) to enhance test-time robustness, typically using AugMix to generate multiple views per image. Methods such as VTE and Zero aggregate predictions across these views to reduce randomness, while MEMO (Zhang, Levine, and Finn 2022), TPT (Shu et al. 2022), and TPS (Sui, Wang, and Yeung-Levy 2025) minimize marginal entropy to enforce consistency across augmented inputs.

3 Challenges

Preliminary CLIP (Radford et al. 2021) is a vision-language model (VLM) with an image encoder $\mathcal{E}_v$ and a text encoder $\mathcal{E}_t$, which aligns images with their corresponding textual descriptions. By pretraining on a large-scale image-text dataset, CLIP is capable of zero-shot prediction. Specifically, for a classification task with C classes, the text encoder $\mathcal{E}_t$ embeds class descriptions (e.g., "`a photo of a {class}`") into normalized text embeddings $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_C \in \mathbb{R}^D$. Given a test image, the image encoder $\mathcal{E}_v$ produces a normalized image feature $\boldsymbol{v}_i \in \mathbb{R}^D$, and the prediction is made by assigning the image to the class with the highest similarity score, i.e., $\hat{y}_i = \arg\max_c \boldsymbol{v}_i^\top \boldsymbol{t}_c$.

Prediction bias from corruptions Although CLIP exhibits strong zero-shot generalization capabilities, its classification accuracy often degrades in the presence of common corruptions (Hendrycks and Dietterich 2019). Corruptions such as Gaussian noise and defocus blur can be encoded into the image embeddings, introducing bias into the representation. When these biases correlate spuriously with text embeddings, they lead to *prediction bias*, where corrupted images are disproportionately assigned to certain classes. To quantify this effect, we evaluate the distribution distance between the ground-truth label distribution and the soft predic-

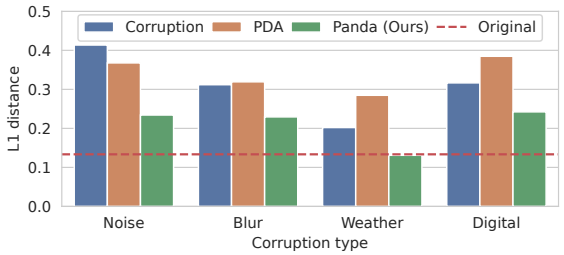

Figure 2: Distribution distance between ground-truth and soft prediction distributions under four corruption categories. *Original* denotes the uncorrupted CIFAR-10 dataset. Larger distribution distance indicates greater prediction bias. Corruptions introduce significant bias that positive data augmentation often fails to mitigate. In contrast, `Panda` effectively reduces this bias. See Figure 7 in Appendix A for results on all 15 corruption types.

tion distribution on CIFAR-10-C (Krizhevsky, Hinton et al. 2009; Hendrycks and Dietterich 2019). A larger distance indicates more severe prediction bias. As shown in Figure 2, each type of corruption significantly increases prediction bias. Prediction bias not only directly reduces classification accuracy but also poses a critical challenge for entropy-based TTA methods. Since these methods treat soft predictions as pseudo-labels, the presence of prediction bias can amplify itself during adaptation and potentially lead to model collapse (Niu et al. 2023; Zhao et al. 2023). Therefore, reducing prediction bias is essential for robust TTA.

PDA fails to alleviate prediction bias Many TTA methods (Zhang, Levine, and Finn 2022; Shu et al. 2022; Sui, Wang, and Yeung-Levy 2025; Döbler et al. 2024; Farina et al. 2024) use positive data augmentation (PDA) (Hendrycks et al. 2020) to improve robustness to image corruptions. These methods generate multiple semantic-preserving views for each test image and average the predictions from high-confidence views. However, as shown in Figure 2, we observe that PDA often fails to alleviate prediction bias and sometimes even amplifies it. This occurs because positive augmentations do not remove the corruption present in the image. In addition, selecting and averaging high-confidence views can unintentionally strengthen the influence of corruption within the image embedding. In comparison, our proposed method `Panda` adopts negative data augmentation (NDA) to reduce prediction bias across all corruption types. In the following section, we describe how we design the NDA process and how its outputs are used to debias image embeddings and adapt the model accordingly.

4 Algorithm

Figure 3 gives an overview of our proposed algorithm `Panda`, which consists of three main steps:

1. For each test batch of B images, we generate $M \ll B$ negative augmentations. (Subsection 4.1)
2. We encode both the test samples and the negative augmentations into image embeddings. The average embedding of the negative augmentations is used to offset each test sample, reducing the influence of corruption. (Subsection 4.2).
3. The debiased features are then passed to any existing TTA method to refine predictions or adapt the model. (Subsection 4.3)

4.1 Negative Data Augmentation

We begin by introducing our negative data augmentation (NDA) strategy. The goal of NDA is to generate images from a batch of test samples that preserve corruption-related information while disrupting object-related content. Specifically, our NDA takes a batch of B images $\{\boldsymbol{x}_i\}_{i=1}^{B}$ as input. Each image of size $H \times W$ is first partitioned into $\frac{H}{H_p} \times \frac{W}{W_p}$ non-overlapping patches of size $H_p \times W_p$. All patches from the batch are collected and randomly shuffled to form a patch pool. From this pool, patches are selected and combined to recompose M *negatively augmented images* $\{\boldsymbol{x}_j^-\}_{j=1}^{M}$, where each negative augmentation image is constructed by combining patches such that every patch is used at most once within the batch, and the size of each image remains $H \times W$. These negatively augmented images are fed into the image encoder $\mathcal{E}_v$ alongside the original batch.

Compared to PDA, our NDA strategy requires significantly fewer augmentations. PDA typically generates K augmentations for each image in the batch (commonly $K = 63$), resulting in a forward pass cost of $K + 1$ times. In contrast, NDA does not need to preserve object information for individual images, allowing all samples in a batch to share a common set of M negative augmentations. In practice, we typically set $M = B/10$, which is much smaller than the batch size B, leading to substantial computational savings.

4.2 Offset

We feed both the original B test images and the M negatively augmented images into the image encoder, obtaining image embeddings $\{\boldsymbol{v}_i = \mathcal{E}_v(\boldsymbol{x}_i)\}_{i=1}^{B}$ for the original inputs and $\{\boldsymbol{n}_j = \mathcal{E}_v(\boldsymbol{x}_j^-)\}_{j=1}^{M}$ for the negatively augmented images. We aggregate the negatively augmented embedding by computing their average:

$$\bar{\boldsymbol{n}} = \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{n}_j, \tag{1}$$

Due to the patch-level shuffling and averaging across multiple negatively augmented images, the resulting embedding $\bar{\boldsymbol{n}}$ contains minimal object-related information while retaining corruption-related characteristics. We use $\bar{\boldsymbol{n}}$ to offset the original image embeddings in order to suppress the corruption components:

$$\boldsymbol{d}_i = \boldsymbol{v}_i - \beta \cdot \bar{\boldsymbol{n}}, \quad i = 1, \cdots, B, \tag{2}$$

where $\beta > 0$ is a hyperparameter controlling the offset ratio. We use the debiased embedding $\{\boldsymbol{d}_i\}_{i=1}^{B}$ as a replacement of the original image embedding $\{\boldsymbol{v}_i\}_{i=1}^{B}$. When not combined

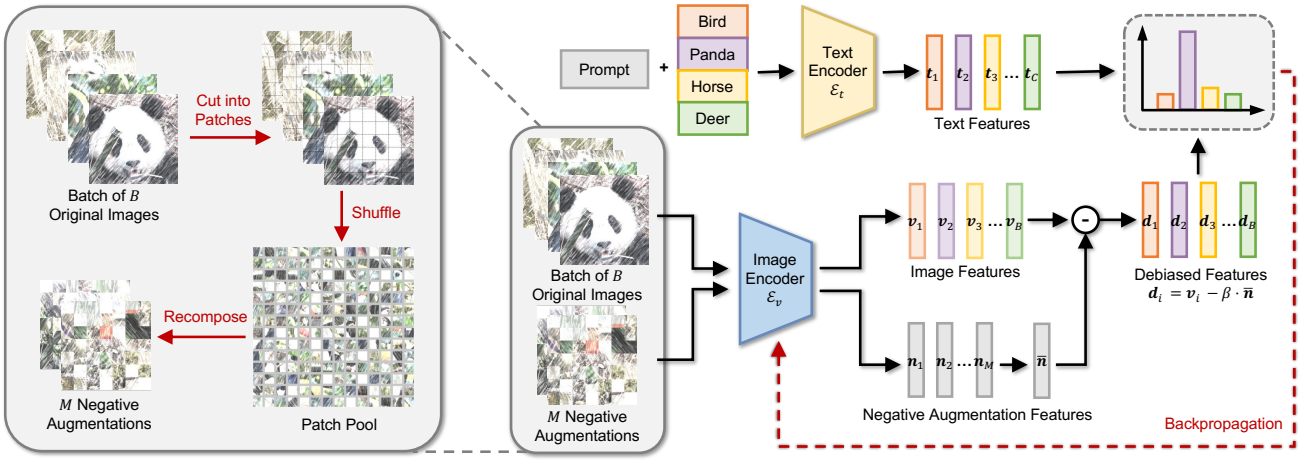

Figure 3: Overview of Panda. Given a batch of B original images, M negatively augmented images are generated by cutting the originals into patches, shuffling, and recomposing. Both original and negative augmented images are encoded by the image encoder. The average of the negative embeddings serves as a corruption prototype and is subtracted from the original embeddings to suppress corruption-related features. Final predictions are obtained by comparing the debiased features with text embeddings.

with other TTA algorithms, the prediction is given by $\hat{y}_i = \arg\max_c \boldsymbol{d}_i^\top \boldsymbol{t}_c$, where $\boldsymbol{t}_c$ is the text embedding for class c.

In Theorem 4.1 below, we use an one-dimensional example to justify the validity of this offset approach. Note that this conclusion can be generalized to high-dimensional settings. The proof and its high-dimensional generalization are provided in Appendix B.

Theorem 4.1 (Offsetting leads to accuracy gain)**.** *Consider a binary classification problem where the input feature v can be decomposed into two independent components: $v = v_{cls} + v_{corr}$, where $v_{cls} \sim \mathcal{N}(0,1)$ denotes the class-relevant component and $v_{corr} \sim \mathcal{N}(0, s^2)$ denotes the corruption-related component, with $s > 0$ representing corruption severity. Let the ground-truth label be $y = \text{sign}(v_{cls})$, and the classifier be $\hat{y} = \text{sign}(v)$. Then the classification accuracy is*

$$\Pr(\text{sign}(v) = y) = \frac{1}{2} + \frac{1}{\pi} \cdot \arctan\left(\frac{1}{s}\right). \tag{3}$$

Now consider a negatively augmented feature $n \sim \mathcal{N}(0, s^2)$ such that the correlation $\rho(n, v_{cls}) = 0$ and $\rho(n, v_{corr}) = r > 0$. Then offsetting v using n yields improved accuracy:

$$\Pr(\text{sign}(v - \beta \cdot n) = y) = \frac{1}{2} + \frac{1}{\pi} \cdot \arctan\left(\frac{1}{s \cdot \sqrt{1 - r^2 + (\beta - r)^2}}\right), \tag{4}$$

which is maximized when $\beta = r$.

Theorem 4.1 shows that when the negative augmentation effectively removes object information and retains part of the corruption characteristics, our offset strategy can suppress the corruption component in the original feature down to a minimum of $\sqrt{1 - r^2}$ times its original magnitude, leading to improved accuracy. This result highlights the effectiveness of using negative augmentation to isolate and suppress corruption in the feature space.

Algorithm 1: Tent + Panda

Input: Test data stream $\{\mathcal{X}_t\}_{t=1}^T$ with $\mathcal{X}_t = \{\boldsymbol{x}_i\}_{i=1}^B$; image encoder $\mathcal{E}_v(\cdot; \boldsymbol{w})$ with parameters $\boldsymbol{w}$ (e.g., LayerNorm); text embeddings $\{\boldsymbol{t}_c\}_{c=1}^C$; learning rate η; patch size $H_p \times W_p$; offset ratio β; number of negative augmentations $M = \lceil B/10 \rceil$

Output: Predictions $\{\hat{y}_i\}_{i=1}^B$ for each batch

for each test batch $\mathcal{X}_t$ **do**
  # Negative data augmentation
  Generate M negative augmentations $\{\boldsymbol{x}_j^-\}_{j=1}^M$ by cuting $\boldsymbol{x}_i \in \mathcal{X}_t$ into patches, shuffling, and recomposing
  # Feature encoding
  $[\boldsymbol{v}_1, \cdots, \boldsymbol{v}_B] \leftarrow \text{normalize}\left(\mathcal{E}_v([\boldsymbol{x}_1, \cdots, \boldsymbol{x}_B]; \boldsymbol{w})\right)$
  $[\boldsymbol{n}_1, \cdots, \boldsymbol{n}_M] \leftarrow \text{normalize}\left(\mathcal{E}_v([\boldsymbol{x}_1^-, \cdots, \boldsymbol{x}_M^-]; \boldsymbol{w})\right)$
  # Feature debiasing
  $\bar{\boldsymbol{n}} \leftarrow \frac{1}{M}\sum_{j=1}^M \boldsymbol{n}_j$
  $\boldsymbol{d}_i \leftarrow \boldsymbol{v}_i - \beta \cdot \bar{\boldsymbol{n}}$, for $i = 1, \cdots, B$
  # Adaptation
  $\text{logits}_i \leftarrow 100 \cdot \boldsymbol{d}_i^\top [\boldsymbol{t}_1, \cdots, \boldsymbol{t}_C]$, for $i = 1, \cdots, B$
  $\mathcal{L} \leftarrow \frac{1}{B}\sum_{i=1}^B \mathcal{H}(\text{logits}_i)$
  $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \cdot \nabla_{\boldsymbol{w}} \mathcal{L}$ *# update parameters of $\mathcal{E}_v$*
  $\hat{y}_i \leftarrow \arg\max_c (\boldsymbol{d}_i^\top \boldsymbol{t}_c)$ for $i = 1, \cdots, B$
Output $\{\hat{y}_i\}_{i=1}^B$

4.3 Combination with TTA Methods

Panda modifies only the forward pass with minimal computational overhead, which makes it compatible with most existing TTA methods. For instance, Algorithm 1 gives an example when Panda is integrated with Tent (Wang et al. 2021). The standard objective minimizes the entropy of predictions based on the original image features $\boldsymbol{v}_i$. In our method, we instead use the debiased features $\boldsymbol{d}_i$ to replace $\boldsymbol{v}_i$ in the forward computation. This substitution reduces bias in the logits, thereby improving both prediction quality and adaptation stability. As shown in our experiments, existing

		CLIP	Tent	ETA	SAR	DeYO	TPT	DMN-ZS	Zero	TPS	BAT
ViT-B/32 on CIFAR-10-C	Baseline	59.0 (0.0)	62.8 (0.1)	64.9 (0.1)	63.3 (0.3)	65.5 (0.0)	62.2 (0.1)	61.6 (0.0)	63.2 (0.0)	63.7 (0.0)	65.7 (0.0)
	+Panda	61.6 (0.0)	71.1 (0.1)	68.3 (0.1)	70.7 (0.2)	67.2 (0.1)	63.5 (0.0)	63.1 (0.1)	64.9 (0.0)	65.6 (0.0)	68.5 (0.0)
	Δ	+2.6	+8.3	+3.4	+7.4	+1.7	+1.3	+1.5	+1.7	+1.9	+2.8
ViT-B/32 on CIFAR-100-C	Baseline	31.8 (0.0)	35.7 (0.1)	40.8 (0.4)	39.1 (0.3)	38.0 (0.3)	32.1 (0.0)	32.2 (0.1)	29.5 (0.1)	33.1 (0.1)	36.5 (0.0)
	+Panda	33.4 (0.0)	38.4 (0.1)	43.3 (0.1)	41.7 (0.2)	42.1 (0.2)	34.3 (0.0)	34.2 (0.1)	33.3 (0.0)	35.1 (0.0)	37.7 (0.0)
	Δ	+1.6	+2.7	+2.5	+2.6	+4.1	+2.2	+2.0	+3.8	+2.0	+1.2
ViT-B/16 on ImageNet-C	Baseline	24.5 (0.0)	25.3 (0.0)	26.5 (0.0)	31.8 (0.2)	29.0 (0.2)	25.2 (0.0)	24.6 (0.0)	24.6 (0.1)	25.1 (0.1)	25.6 (0.0)
	+Panda	26.2 (0.0)	28.2 (0.1)	27.9 (0.0)	32.4 (0.2)	31.2 (0.2)	27.1 (0.0)	26.2 (0.0)	27.1 (0.0)	27.2 (0.0)	27.2 (0.1)
	Δ	+1.7	+2.9	+1.4	+0.6	+2.2	+1.9	+1.6	+2.5	+2.1	+1.6

Table 1: Comparison of accuracy (mean (s.d.) %) between the single baseline method and the baseline integrated with Panda. Only the average accuracy over 15 corruption types is reported; full per-corruption results are deferred to Appendix C.

TTA algorithms can achieve better performance when combined with Panda.

5 Experiments

In this section, we conduct experiments to investigate the following research questions:

- **RQ1**: Can Panda enhance the performance of existing TTA methods significantly in diverse datasets?
- **RQ2**: Compared to PDA methods, does Panda achieve better performance with higher efficiency?
- **RQ3**: Does Panda effectively alleviate prediction bias?

Setup We conduct experiments on three widely used corruption benchmarks: CIFAR-10-C, CIFAR-100-C, and ImageNet-C, each containing 15 types of common corruptions. Following standard practice (Maharana et al. 2025; Bao, Deng, and He 2025), we evaluate our method at corruption severity level 5. For the backbone, we use ViT-B/32 for CIFAR-10-C and CIFAR-100-C, and ViT-B/16 for ImageNet-C. We adopt a default batch size of 100 for adaptation and use a fixed prompt template ("a photo of a {class}") for the text encoder. All images from the corruption datasets are resized to 224×224 to match the input resolution required by the ViT backbone. We use a default patch size of $H_p = W_p = 32$, meaning each original image is partitioned into 7×7 non-overlapping patches (i.e., $224/32 = 7$ per dimension) during negative augmentation.

Baselines Besides CLIP (Radford et al. 2021), we assess the effectiveness of Panda in conjunction with nine different TTA baselines, covering diverse adaptation types and models designs. For general TTA methods, we consider Tent (Wang et al. 2021), ETA (Niu et al. 2022), SAR (Niu et al. 2023), and DeYO (Lee et al. 2024). For CLIP-specific TTA methods, we compare both model-adaptive approaches including TPT (Shu et al. 2022), TPS (Sui, Wang, and Yeung-Levy 2025), and BAT (Maharana et al. 2025), and training-free methods including DMN-ZS (Zhang et al. 2024) and Zero (Farina et al. 2024). Among them, TPT, TPS, and Zero adopt AugMix (Hendrycks et al. 2020) as their positive data augmentation strategy. For each baseline, in addition to evaluating the method itself, we also report a combined version that integrates Panda by replacing the original image features $\boldsymbol{v}_i$ with the debiased features $\boldsymbol{d}_i = \boldsymbol{v}_i - \beta \cdot \bar{\boldsymbol{n}}$ during both logit computation and adaptation. Unless otherwise specified, all hyperparameters and experimental protocols are optimized using the original baseline alone and kept identical for its Panda-augmented counterpart. Please refer to Appendix C for full implementation details and hyperparameter values.

Method	CIFAR-10-C	CIFAR-100-C	ImageNet-C
TPT	62.2 (0.1)	32.1 (0.0)	25.2 (0.0)
Zero	63.2 (0.0)	29.5 (0.1)	24.6 (0.1)
TPS	63.7 (0.0)	33.1 (0.1)	25.1 (0.1)
Panda	**71.1** (0.1)	**38.4** (0.1)	**28.2** (0.1)

Table 2: Comparison of mean accuracy (%) between Panda and PDA methods (TPT, Zero, and TPS). Only the average accuracy over 15 corruption types is reported; full per-corruption results are deferred to Appendix C.

Combination with existing TTA methods (RQ1) We evaluate the effectiveness of Panda by integrating it into a wide range of existing TTA methods across three corruption benchmarks. As shown in Table 1, Panda consistently improves the performance of all baselines across all datasets, demonstrating its strong compatibility with existing methods and its ability to effectively leverage high-quality negative augmentations to suppress spurious features in original images under various settings.

On CIFAR-10-C, Panda brings substantial gains to general TTA methods such as Tent (+8.3%), ETA (+3.4%), and SAR (+7.4%), with an average improvement of +3.3% across all baselines. On CIFAR-100-C, Panda achieves an average improvement of +2.2% over all baselines, with the highest gain of +4.1%. On ImageNet-C, it brings an average improvement of +2.0%, with a maximum gain of +2.9%.

Notably, Panda can also be integrated with PDA methods and achieve better performance, as evidenced by improvements on TPT (+2.2%), TPS (+3.8%), and Zero (+2.0%) on CIFAR-100-C. In addition, Panda can also be integrated with the NDA method DeYO to enhance its ability, which as shown by improvements on DeYO (+4.1%) on

Method	CLIP	Tent	ETA	SAR	DeYO	TPT	DMN	Zero	TPS	BAT
Baseline Time	17s	25s	21s	31s	27s	22min21s	22s	8min51s	9min32s	28s
+Panda Time	18s	27s	23s	34s	28s	22min39s	23s	8min55s	9min37s	30s
Overhead	5.9%	8.0%	9.5%	9.7%	3.7%	1.3%	4.5%	0.8%	0.9%	7.2%

Table 3: Comparison of testing time with baselines and baselines+Panda for ViT-B/32 on CIFAR-10.

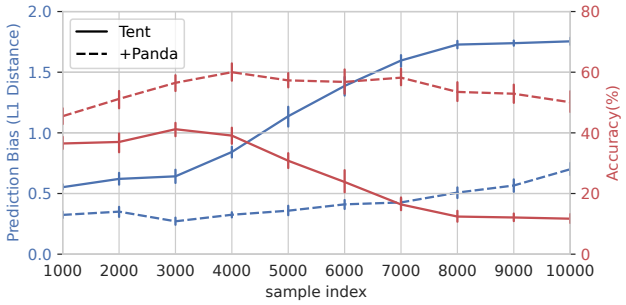

Figure 4: Prediction bias and accuracy (%) measured across the test stream, divided into 10 consecutive chunks (each of 1,000 samples).

Method	CIFAR-10-C	CIFAR-100-C	ImageNet-C
CLIP	59.0 (0.0)	31.8 (0.0)	24.5 (0.0)
Select & weight	65.5 (0.0)	38.0 (0.3)	29.0 (0.2)
Offset (Ours)	**68.3** (0.0)	**43.3** (0.1)	**29.4** (0.0)

Table 4: Comparison of mean accuracy (%) between for negative augmentation strategies in DeYO and Panda. Selecting and weighting testing samples using NDA is from DeYO and offseting bias features is from Panda. Only the average accuracy over 15 corruption types is reported; full per-corruption results are deferred to Appendix C.

CIFAR-100-C.

Compared with PDA methods (RQ2) Beyond the distributional distance analysis, we conduct controlled experiments under identical settings to compare Panda with positive data augmentation methods, and observe from Table 2 that Panda significantly outperforms all PDA baselines (TPT (Shu et al. 2022), Zero (Farina et al. 2024), and TPS (Sui, Wang, and Yeung-Levy 2025)) by notably reducing prediction bias and achieving higher accuracy.

Efficiency (RQ2) We evaluate the computational overhead introduced by integrating Panda by comparing runtime of baseline methods with and without Panda. From Table 3, incorporating Panda results in less than a 10% increase in runtime compared to their standalone counterparts, which demonstrates that Panda achieves performance improvements with minimal efficiency overhead. Moreover, in contrast to positive data augmentation methods such as TPS, TPT, and Zero that generate numerous independent augmented views for each test sample, Panda with only a small number of negative augmentations consequently achieves substantially lower computational cost and runtime.

Alleviate prediction bias (RQ3) In Figure 2 and Figure 7 in Appendix A, we have verified that applying Panda alone can effectively reduce the model's prediction bias. Here we further investigate whether Panda can alleviate prediction bias when combined with other TTA methods. To this end, we choose Tent (Wang et al. 2021) as a baseline, as it is known to suffer from prediction bias (Niu et al. 2023). We conduct experiments on the gaussian noise corruption from CIFAR-10-C. Fixing the data order, we divide the 10,000 test samples into 10 consecutive chunks (1–1000, 1001–2000, ...). For each chunk, we compute (1) the prediction bias, measured as the L1 distance between the ground-truth label distribution and the soft prediction distribution, and (2) the classification accuracy. As shown in Figure 4, Tent gradually accumulates prediction bias as more test samples are processed. As a result, its accuracy improves slightly only at the beginning but quickly degrades, eventually leading to model collapse. In contrast, Tent + Panda maintains consistently lower prediction bias and achieves substantially higher accuracy throughout the entire test stream. These results highlight that Panda not only alleviates prediction bias in a static sense but also mitigates its accumulation throughout the test-time adaptation process, leading to improved performance.

Comparison of negative augmentation strategies with other NDA methods To evaluate the effectiveness of the negative data augmentation used in Panda, we compare it with a strong NDA-based baseline, DeYO (Lee et al. 2024). DeYO estimates prediction confidence by measuring the discrepancy between predictions on original and negatively augmented images, and uses this confidence to guide sample selection and weighting during adaptation. For a fair comparison, we remove both the NDA-relevant components from DeYO, and instead adopt the NDA generation and offset mechanism used in Panda. As shown in Table 4, the negative augmentation strategy employed by Panda achieves superior performance. This result demonstrates that Panda produces higher-quality negative augmentations that more effectively suppress prediction bias on corrupted data, outperforming existing state-of-the-art NDA approaches in test-time adaptation.

Ratio between B and M B denotes the batch size and M represents the number of negative augmentation images generated per batch. We investigate whether the performance of Panda degrades significantly as the M/B ratio decreases. As shown in Figure 5 (left), Panda maintains stable and strong performance across different M/B settings. The con-

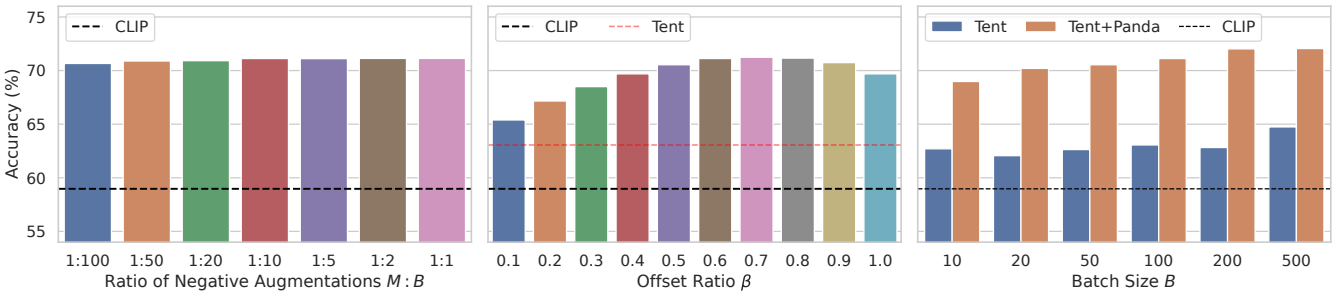

Figure 5: Sensitivity analysis of Panda. Left: accuracy under different ratios of $M : B$, where M is the number of negative augmentations per batch and B is the batch size. Middle: accuracy across a range of offset ratios β used in patch translation. Right: accuracy under varying batch sizes, comparing Tent and Tent+Panda.

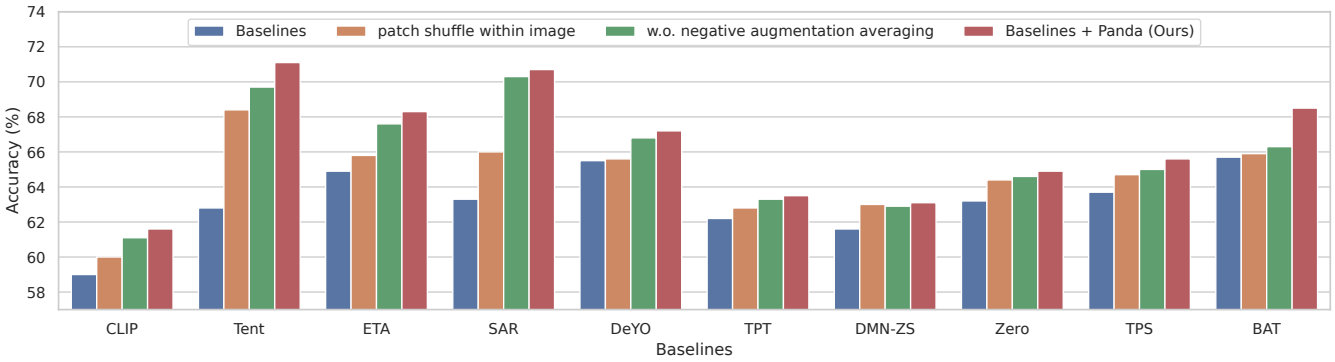

Figure 6: Ablation study comparing the full version of Panda with its decomposed variants across different TTA baselines. Variants include disabling Panda, shuffling patches within individual images, and removing augmentation feature averaging.

sistently high performance under extremely low M/B ratios indicates that generated negative augmentation images exhibit strong information-sharing capacity, allowing a small number of augmentations to benefit a large number of samples effectively.

Hyperparameter sensitivity We conduct sensitivity analyses of Panda with respect to the offset ratio β, batch size, and learning rate. As shown in Figure 5 (middle), Panda consistently yields significantly better performance across a wide range of offset ratio β. This shows that Panda can consistently enhance the performance of baselines across a wide range of offset ratio β, without relying on a specific value. Moreover, as illustrated in Figure 5 (right), when combined with Panda, Tent maintains a consistently large performance gain over the original Tent baseline across different batch sizes. This indicates that Panda can effectively enhance the performance of the base method under varying batch size settings. Detailed results for other baselines under different learning rates are provided in Appendix C.

Ablation study We conduct an ablation study by decomposing components of Panda. Specifically, we compare the full version of Panda with several variants: (1) disabling Panda entirely (2) performing negative augmentation based on the image itself rather than across the whole batch of images, and (3) full version of Panda without the averaging of negative augmentation features. As shown in Figure 6, results across all baselines consistently demonstrate that these ablated variants perform significantly worse than the full version of Panda. This demonstrates that batch-wide negative augmentation generation and feature averaging are both essential for Panda's performance gains.

6 Conclusion

In this work, we introduce Panda, a novel test-time adaptation method that leverages negative data augmentation to mitigate prediction bias caused by image corruptions. Unlike traditional positive augmentation strategies, Panda generates negative augmentations by disrupting object semantics through patch shuffling, effectively preserving corruption-specific characteristics while suppressing object-relevant features. By aggregating features from negatively augmented images, our approach offsets the corruption-induced bias in test samples and significantly reduces computational overhead by enabling shared augmentations within each batch. Extensive experiments on standard corruption benchmarks demonstrate that Panda consistently outperforms positive data augmentation methods and robustly enhances the performance of various TTA frameworks. Our results highlight the practical effectiveness and efficiency of negative data augmentation for robust vision-language model adaptation.

Acknowledgements

This work is supported by National Science Foundation under Award No. IIS-2416070, IIS-2117902. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

References

Bao, W.; Deng, R.; and He, J. 2025. Mint: A Simple Test-Time Adaptation of Vision-Language Models against Common Corruptions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Bao, W.; Deng, R.; Qiu, R.; Wei, T.; Tong, H.; and He, J. 2025a. Latte: Collaborative Test-Time Adaptation of Vision-Language Models in Federated Learning. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, Hawaii, USA, October 19-23, 2025*. IEEE.

Bao, W.; Zeng, Z.; Liu, Z.; Tong, H.; and He, J. 2025b. Matcha: Mitigating Graph Structure Shifts with Test-Time Adaptation. In *13th International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Devries, T.; and Taylor, G. W. 2017. Improved Regularization of Convolutional Neural Networks with Cutout. *CoRR*, abs/1708.04552.

Döbler, M.; Marsden, R. A.; Raichle, T.; and Yang, B. 2024. A Lost Opportunity for Vision-Language Models: A Comparative Study of Online Test-time Adaptation for Vision-Language Models. *CoRR*, abs/2405.14977.

Farina, M.; Franchi, G.; Iacca, G.; Mancini, M.; and Ricci, E. 2024. Frustratingly Easy Test-Time Adaptation of Vision-Language Models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Dec 10–15, 2024, Vancouver, BC, Canada*.

Hendrycks, D.; and Dietterich, T. G. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *7th International Conference on Learning Representations, ICLR 2019, May 6–9, 2019, New Orleans, LA, USA*. OpenReview.net.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2020. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Karmanov, A.; Guan, D.; Lu, S.; El-Saddik, A.; and Xing, E. P. 2024. Efficient Test-Time Adaptation of Vision-Language Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Jun 16–22, 2024, Seattle, WA, USA*, 14162–14171. IEEE.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. -.

Lee, J.; Jung, D.; Lee, S.; Park, J.; Shin, J.; Hwang, U.; and Yoon, S. 2024. Entropy is not Enough for Test-Time Adaptation: From the Perspective of Disentangled Factors. In *12th International Conference on Learning Representations, ICLR 2024, May 7–11, 2024, Vienna, Austria*. OpenReview.net.

Liang, J.; He, R.; and Tan, T. 2024. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 31–64.

Maharana, S. K.; Zhang, B.; Karlinsky, L.; Feris, R.; and Guo, Y. 2025. BATCLIP: Bimodal Online Test-Time Adaptation for CLIP. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025*. IEEE.

Niu, S.; Wu, J.; Zhang, Y.; Chen, Y.; Zheng, S.; Zhao, P.; and Tan, M. 2022. Efficient Test-Time Model Adaptation without Forgetting. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 16888–16905. PMLR.

Niu, S.; Wu, J.; Zhang, Y.; Wen, Z.; Chen, Y.; Zhao, P.; and Tan, M. 2023. Towards Stable Test-time Adaptation in Dynamic Wild World. In *11th International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Patashnik, O.; Wu, Z.; Shechtman, E.; Cohen-Or, D.; and Lischinski, D. 2021. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Oct 10–17, 2021, Montreal, QC, Canada*, 2065–2074. IEEE.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, Jul 18–24 2021, Virtual*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Rao, Y.; Zhao, W.; Chen, G.; Tang, Y.; Zhu, Z.; Huang, G.; Zhou, J.; and Lu, J. 2022. DenseCLIP: Language-Guided Dense Prediction with Context-Aware Prompting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, Jun 18–24, 2022, New Orleans, LA, USA*, 18061–18070. IEEE.

Shu, M.; Nie, W.; Huang, D.-A.; Yu, Z.; Goldstein, T.; Anandkumar, A.; and Xiao, C. 2022. Test-Time Prompt Tuning for Zero-Shot Generalization in Vision-Language Models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, Nov 28–Dec 9, 2022, New Orleans, LA, USA*.

Sui, E.; Wang, X.; and Yeung-Levy, S. 2025. Just Shift It: Test-Time Prototype Shifting for Zero-Shot Generalization with Vision-Language Models. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2025, Feb 26–Mar 6, 2025, Tucson, AZ, USA*, 825–835. IEEE.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B. A.; and Darrell, T. 2021. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Xiao, Z.; and Snoek, C. G. M. 2024. Beyond Model Adaptation at Test Time: A Survey. *CoRR*, abs/2411.03687.

Yun, S.; Han, D.; Chun, S.; Oh, S. J.; Yoo, Y.; and Choe, J. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 6022–6031. IEEE.

Zhang, H.; Cissé, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Zhang, M.; Levine, S.; and Finn, C. 2022. MEMO: Test Time Robustness via Adaptation and Augmentation. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, Nov 28–Dec 9, 2022, New Orleans, LA, USA*.

Zhang, Y.; Zhu, W.; Tang, H.; Ma, Z.; Zhou, K.; and Zhang, L. 2024. Dual Memory Networks: A Versatile Adaptation Approach for Vision-Language Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Jun 16–22, 2024, Seattle, WA, USA*, 28718–28728. IEEE.

Zhao, H.; Liu, Y.; Alahi, A.; and Lin, T. 2023. On Pitfalls of Test-Time Adaptation. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 42058–42080. PMLR.

Zhong, Y.; Yang, J.; Zhang, P.; Li, C.; Codella, N.; Li, L. H.; Zhou, L.; Dai, X.; Yuan, L.; Li, Y.; and Gao, J. 2022. RegionCLIP: Region-based Language-Image Pretraining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, Jun 18–24, 2022, New Orleans, LA, USA*, 16772–16782. IEEE.