Mint: A Simple Test-Time Adaptation of Vision-Language Models against Common Corruptions

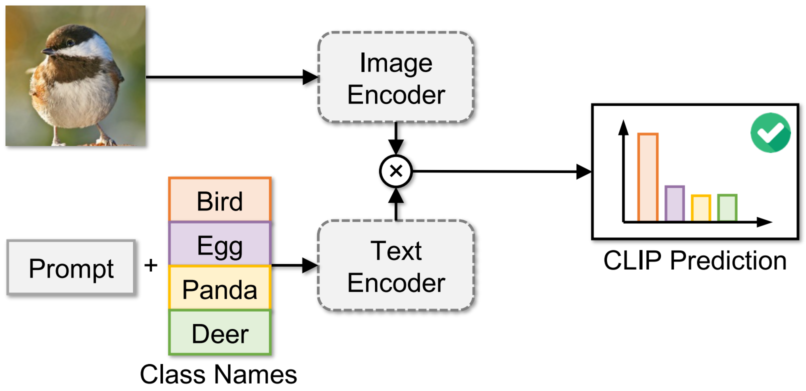
Wenxuan Bao*, Ruxi Deng*, Jingrui He

University of Illinois Urbana-Champaign

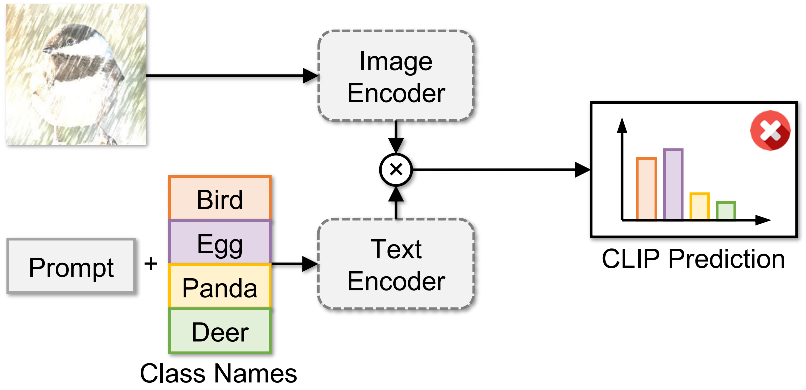
{wbao4,ruxid2,jingrui}@illinois.edu

Background: Vision-Language Model

- **Vision-Language Models** (VLMs) are pre-trained on (image, caption) datasets and possess zero-shot prediction ability.
  - Compare image embedding to text embeddings of "`a photo of a {class}`".
- VLMs are vulnerable to image corruptions.
  - Noise, blur, weather, digital transforms…
- **Test-Time Adaptation** (TTA)
  - Adapt to an unlabeled target domain, without accessing the source data.
  - Improve the performance of VLMs on downstream tasks with distribution shifts.



Zero-shot Prediction



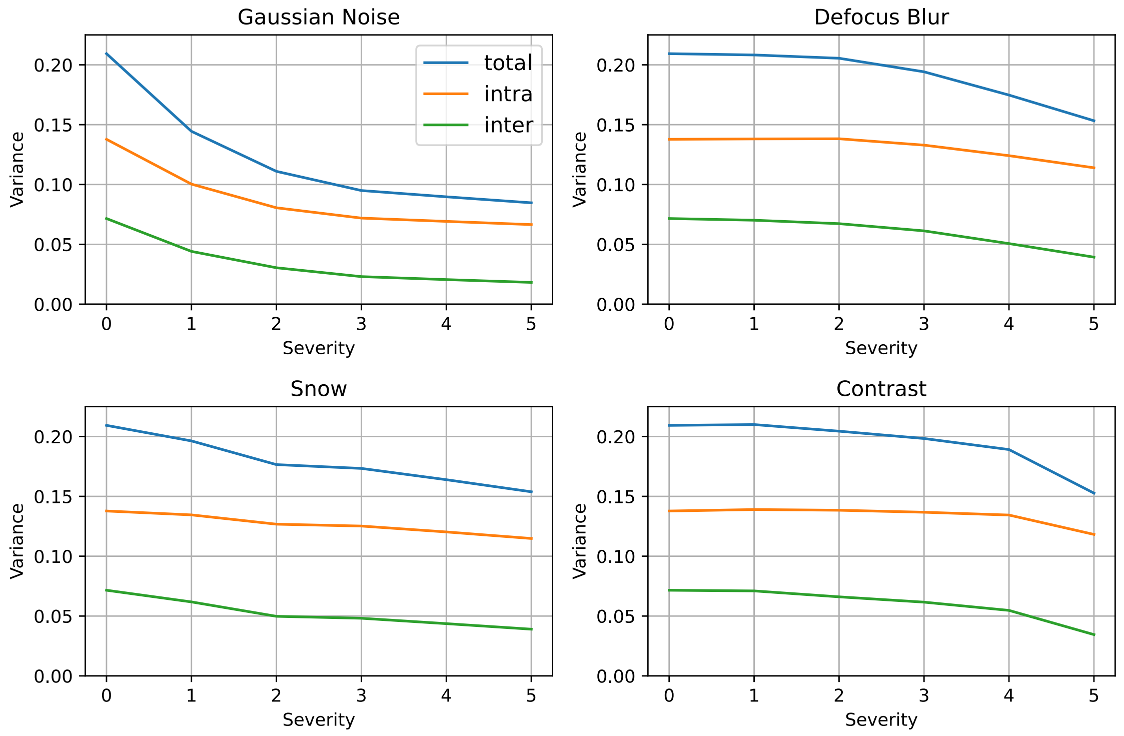
Vulnerability to Common Corruptions

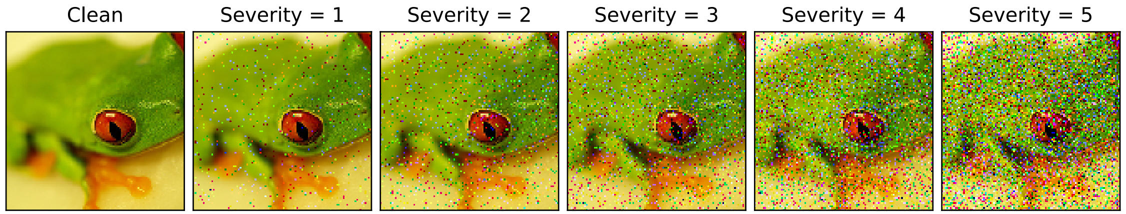
[1] Alec Radford, et al. Learning Transferable Visual Models From Natural Language Supervision. ICML 2021.
[2] Dan Hendrycks, Thomas G. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. ICLR 2019.

How Does Corruption Affect Image Embedding?

- **Variance collapse.** All types of variances decrease under more severe corruption.
  - *GT-intra variance*: embedding variance within the same class,
  - *GT-inter variance*: difference among different classes.

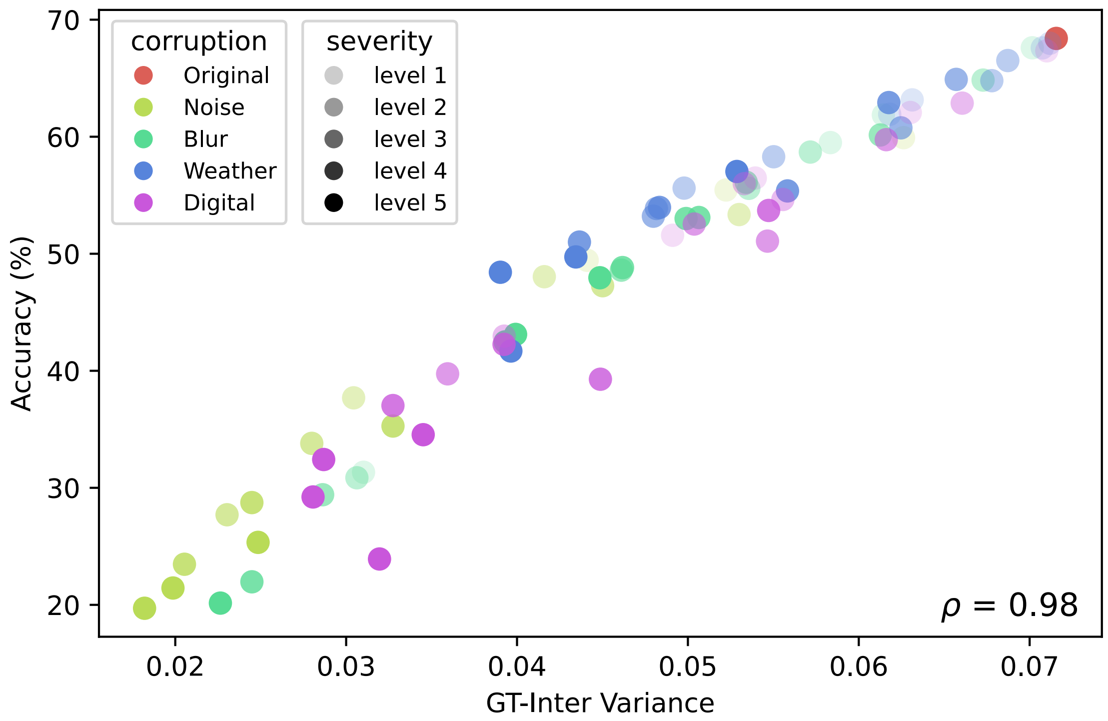
- **Theorem 3.1** (informal)**.** When corruption severity increases,
  - GT-inter variance strictly decreases,
  - GT-intra variance decreases with *large structured distribution shifts*.





Maximizing Inter Variance and Improving Accuracy

- GT-inter variance is highly correlated with accuracy.



- Can we maximize inter-variance to improve accuracy?
  - Ground truth (GT) label y_i not available
  - Use pseudo-label (PL) $\hat{y}_i$ instead, i.e., the zero-shot prediction.

- **Theorem 3.2** (informal)**.** Updating LayerNorm parameters reweights features. Maximizing PL-inter variance can
  - Suppresses structural distribution shifts,
  - Enhances task-relevant features.

Mint: Maximize PL-Inter Variance On-the-Fly

- PL-inter variance over whole test set $\mathbb{D}^{\text{te}}$ can be decomposed as

$$\mathcal{V}_{\text{inter}}^{\text{PL}}(\mathbb{D}^{\text{te}}) = \frac{1}{C}\sum_{c=1}^{C} \|\tilde{\mathbf{z}}_c - \tilde{\mathbf{z}}\|_2^2 = \frac{1}{C}\sum_{c=1}^{C} \frac{\sum_{i=1}^{N} \hat{y}_{ic}\|\mathbf{z}_i - \tilde{\mathbf{z}}\|_2^2}{\sum_{i=1}^{N} \hat{y}_{ic}} - \frac{1}{C}\sum_{c=1}^{C} \frac{\sum_{i=1}^{N} \hat{y}_{ic}\|\mathbf{z}_i - \tilde{\mathbf{z}}_c\|_2^2}{\sum_{i=1}^{N} \hat{y}_{ic}}$$

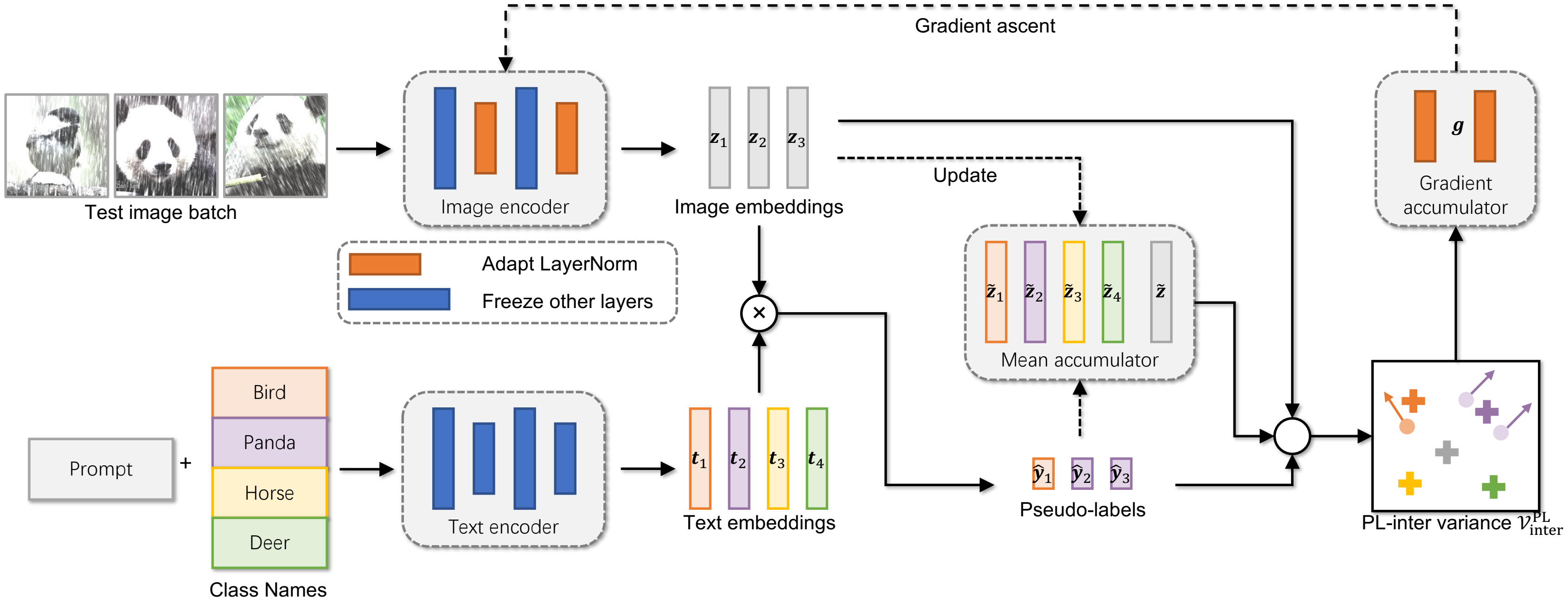
  - $\mathbf{z}_i$: Image embedding of sample i.
  - $\tilde{\mathbf{z}}_c$: Mean embedding of samples with pseudo-label c.
  - $\tilde{\mathbf{z}}$: Mean embedding of all samples.

- Estimated PL-inter variance on a small batch $\mathbb{B}$

$$\mathcal{V}_{\text{inter}}^{\text{PL}}(\mathbb{B}) = \frac{1}{C}\sum_{c=1}^{C} \frac{\sum_{i=1}^{N} \hat{y}_{ic}\|\mathbf{z}_i - \tilde{\mathbf{z}}\|_2^2}{\sum_{i=1}^{N} \hat{y}_{ic}} - \frac{1}{C}\sum_{c=1}^{C} \frac{\sum_{i=1}^{N} \hat{y}_{ic}\|\mathbf{z}_i - \tilde{\mathbf{z}}_c\|_2^2}{\sum_{i=1}^{N} \hat{y}_{ic}}$$

  - $\mathbf{z}_i$: Generated by **current batch**, support backprop.
  - $\tilde{\mathbf{z}}, \tilde{\mathbf{z}}_c$: Estimated by **all seen batches**, do not support backprop, but provide more precise estimation.

Mint: Adaptation with Two Accumulators



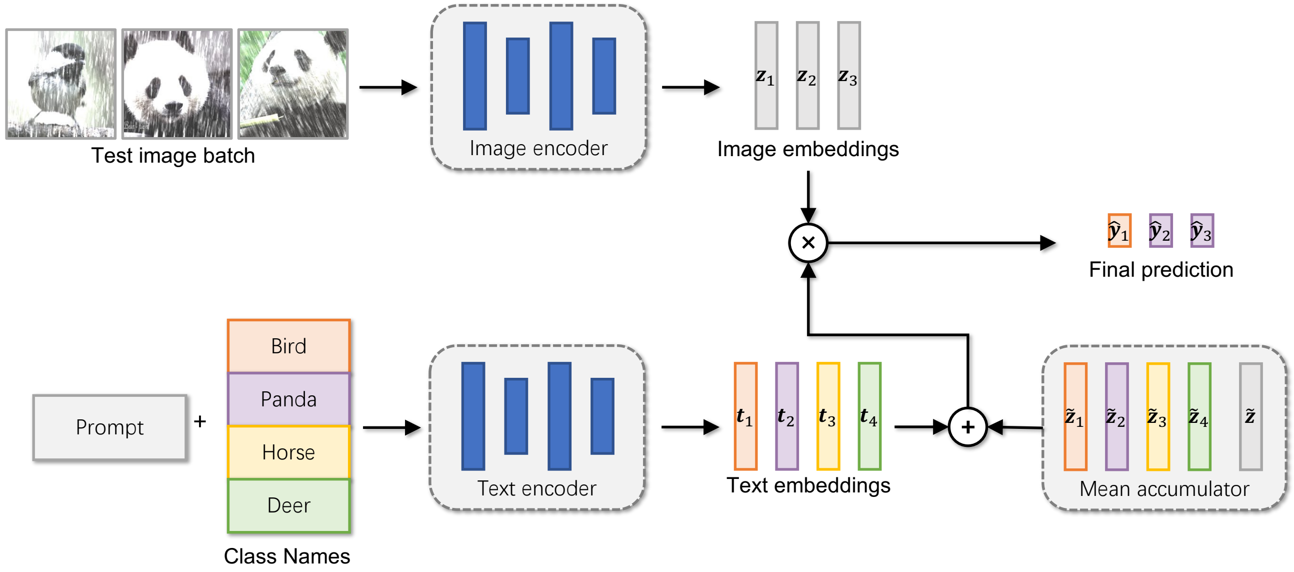
- **Mean accumulator**:
  Moving estimation of $\tilde{\boldsymbol{z}}$ and each $\tilde{\boldsymbol{z}}_c$

$$\tilde{\boldsymbol{z}} \leftarrow \frac{K}{K+1}\tilde{\boldsymbol{z}} + \frac{1}{K+1}\boldsymbol{z}_i, \quad \tilde{\boldsymbol{z}}_{\hat{y}_i} \leftarrow \frac{K_{\hat{y}_i}}{K_{\hat{y}_i}+1}\tilde{\boldsymbol{z}}_{\hat{y}_i} + \frac{1}{K_{\hat{y}_i}+1}\boldsymbol{z}_i.$$

- **Gradient accumulator:**
  Averaging historical gradients to update

$$\bar{\boldsymbol{g}} \leftarrow \frac{b-1}{b}\bar{\boldsymbol{g}} + \frac{1}{b}\boldsymbol{g}_b.$$

Mint: Inference



- Cumulative class means can also be leveraged to adjust the text embedding:

$$\tilde{\boldsymbol{t}}_c \leftarrow \text{normalize}\left(\frac{K_{\text{prior}}}{K_{\text{prior}}+K}\boldsymbol{t}_c + \frac{K}{K_{\text{prior}}+K}\tilde{\boldsymbol{z}}_c\right),$$

where K_{prior} is a hyperparameter controlling the strength of prior. The final prediction is:

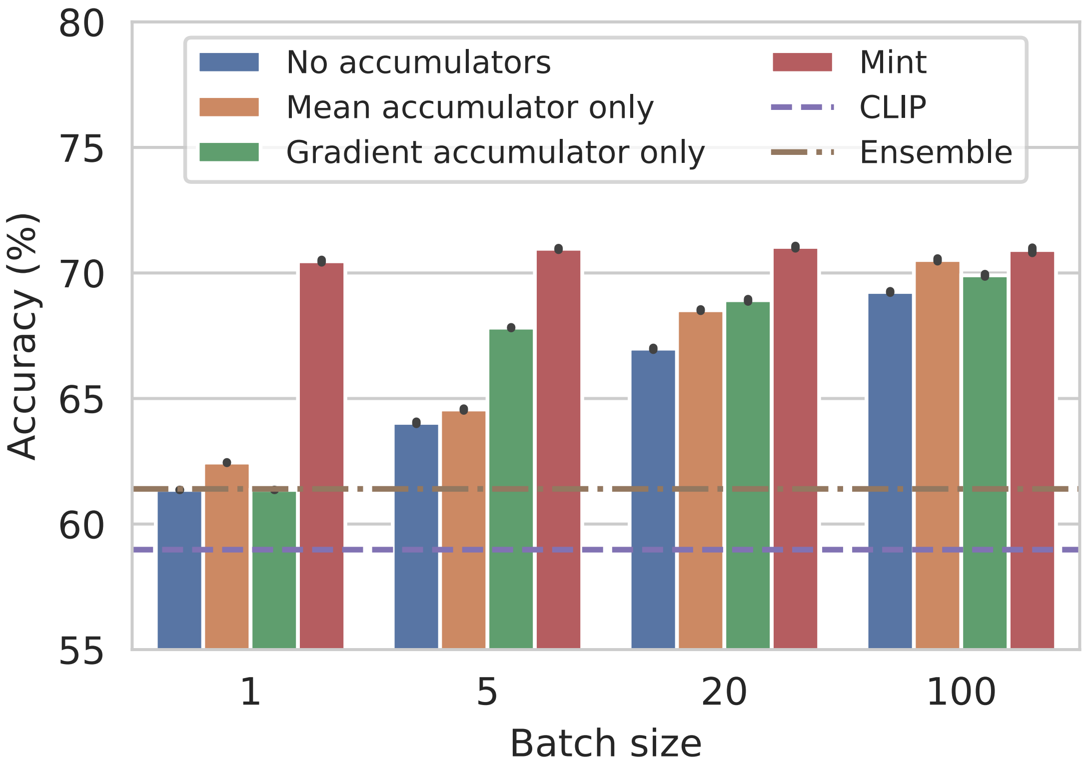
$$\text{argmax}_y \boldsymbol{z}_i^\top \tilde{\boldsymbol{t}}_y$$

Experiments: Effectiveness

Mint has strong performance compared to various types of TTA baselines.

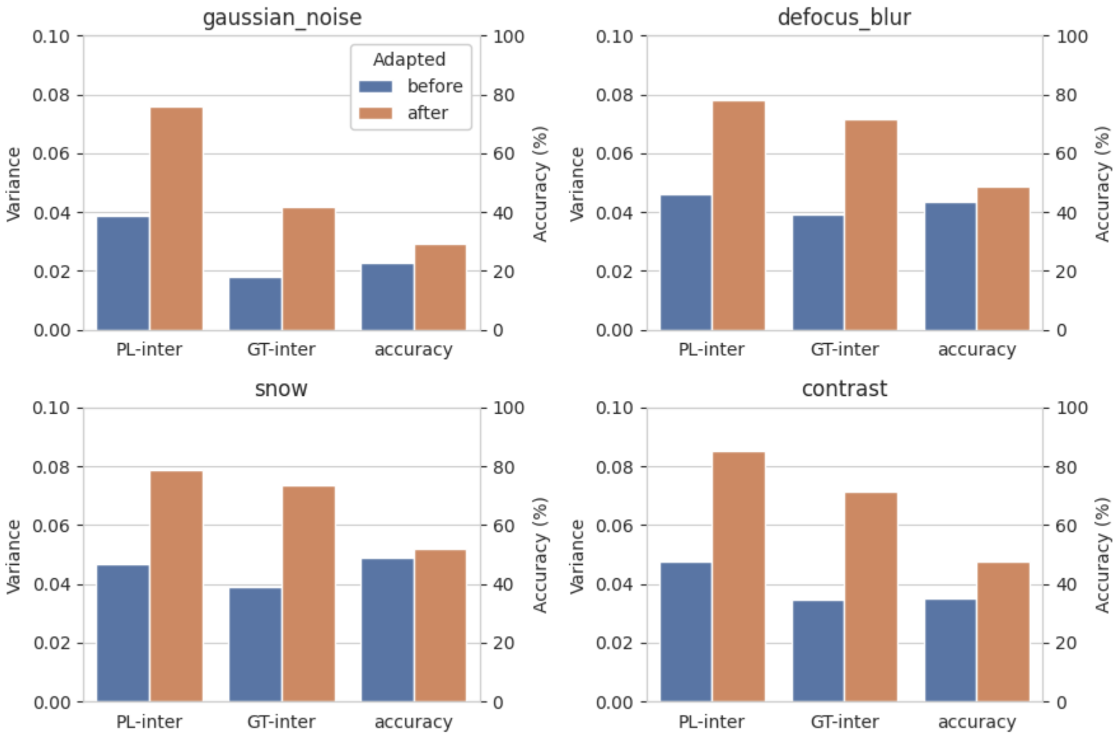
Method	CIFAR-10-C (ViT-B/32)	CIFAR-100-C (ViT-B/16)	ImageNet-C (ViT-L/14)
CLIP	59.0	35.8	39.6
Ensemble	61.4	37.1	41.6
TPT	62.8	36.0	41.1
TDA	62.4	38.4	42.3
DMN-ZS	59.2	38.5	41.7
VTE	65.3	36.6	40.2
Zero	65.3	36.8	40.9
WATT-S	67.1	41.9	43.9
TPS	66.4	38.6	43.1
CLIPArTT	64.4	40.7	40.7
Mint	**71.0**	**44.1**	**47.0**

Two accumulators ensure Mint's robust performance across different batch sizes.



Experiments: Variance Collapse Alleviation & Efficiency

Mint alleviates variance collapse.



Mint is more efficient than most of the TTA baselines, except the training-free ones.

Method	Testing Time	Accuracy (%)	Gain (%)
CLIP	21s	35.8	-
TPT	23m 21s	36.0	+0.2
VTE	9m45s	36.6	+0.8
Zero	9m50s	36.8	+1.0
Ensemble	21s	37.1	+1.3
TDA	33s	38.4	+2.6
DMN-ZS	30s	38.5	+2.7
TPS	9m 58s	38.6	+2.8
CLIPArTT	7m 40s	40.7	+4.9
WATT-S	50m 20s	41.9	+6.1
Mint	1m 07s	44.1	+8.3