BATCLIP: Bimodal Online Test-Time Adaptation for CLIP

Supplementary Material

In this work, we study the problem of **online** test-time adaptation (TTA) of CLIP [18] towards common image corruptions and propose improved schemes for increasing the robustness of CLIP. In addition, to demonstrate the broad impact of our proposed approach, we evaluate on common domain generalization datasets [7]-OfficeHome [23], PACS [12], VLCS [6], and Terra Incognita [2]. We put forward a *bimodal* domain adaptation scheme, at test-time, wherein we exploit the shared feature space of CLIP. In essence, leaning towards a more effective multimodal learning and adaptation method, we propose loss components that improve alignment between the class-specific visual prototype and corresponding text features via maximizing the projection. We also increase the cosine distance between the class prototypes to enhance discrimination between visual features. In this Supplementary, we provide additional insights and experimental results, that has been organized as follows,

1. Section 0.1 offers a detailed discussion of the standard common corruption datasets [9] used, supplemented with visual illustrations.
2. To ensure full transparency, we outline the implementation details of all the methods in Section 0.2, including those for prior TTA approaches [15, 19, 24, 25] adapted for CLIP. We also discuss details of the adapted version of WATT [16] - WATT-P* and WATT-S* for online TTA, and other details of experiments run on the domain generalization datasets.
3. Section 0.3 presents further results and analysis:
   - In subsection 0.3.1, we explore the limitations of zero-shot CLIP when using ViT-B/32 and ViT-L/14 backbones under increasing image corruption severity.
   - In Subsection 0.3.2, we report the main online TTA results on CIFAR-10C, CIFAR-100C, and ImageNet-C with a ViT-B/32 backbone.
   - We study the effect of different prompt templates on BATCLIP in Subsection 0.3.4. Subsection 0.3.5 discusses the ablation of $\mathcal{L}_{pm}$ which is responsible for updating the text encoder.
   - In subsections 0.3.3 and 0.3.8, we present a detailed loss ablation study and the post-adaptation zero-shot generalization on source test sets, respectively — essentially evaluating *catastrophic forgetting* of BATCLIP.
   - Lastly, we include task-wise t-SNE visualizations in subsection 0.3.9 for CIFAR-10C and CIFAR-100C, comparing our method against zero-shot CLIP (ViT-B/16), to illustrate the effectiveness of BATCLIP.

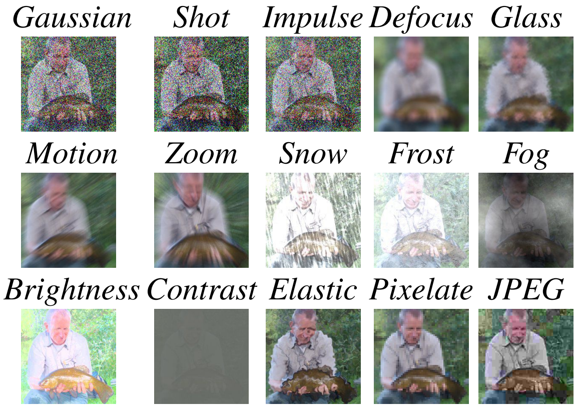

Figure 1. We provide visualizations of an image from ImageNet-C [9] for different corruption types, at an image severity level of 5.

0.1. Datasets

We employ the **CIFAR-10C**, **CIFAR-100C**, and **ImageNet-C** datasets, for our experiments, as introduced by [9]. Each dataset includes 15 distinct types of image corruptions, referred to as tasks in a test-time adaptation setting, applied to the test sets of CIFAR10, CIFAR100 [11], and ImageNet [3]. These corruptions are applied at 5 different severity levels, ranging from mild to severe. For each task, CIFAR-10C and CIFAR-100C contain 10,000 test samples, whereas ImageNet-C has 5000 samples.

The image corruptions are categorized into four primary groups: noise, blur, weather, and digital distortions. Noise-based corruptions include *Gaussian*, *Shot*, and *Impulse* noise, which introduce random pixel-level variations. The blur category encompasses *Defocus*, *Glass*, *Motion*, and *Zoom* blur effects, all of which simulate different types of distorted imagery. Weather-related corruptions, such as *Snow*, *Frost*, and *Fog*, replicate environmental conditions that obscure image details. Lastly, digital distortions include effects like *Brightness*, *Contrast*, *Elastic Transform*, *Pixelate*, and *JPEG* compression, which reflect various forms of post-processing or compression artifacts that degrade image quality.

These corruption types, as proposed by [9], provide a comprehensive framework for assessing model robustness, which has been and is still being studied [10, 13, 14, 17, 21]. Their ability to emulate real-world image degradation scenarios is advantageous, allowing for a more realistic evaluation of a model's robustness. We provide corruption visualizations, via an image example, in Figure 1. We urge the readers to check out [9] for further inspection.

0.2. Implementation Details

In this section, we summarize the implementation details of all the baseline methods that have been mentioned in the main paper, including ours. We build our approach on the standard benchmark code base [1] that also houses the hyperparameters and training details of all the prior TTA methods. CLIP-like models are used as provided by OpenCLIP. Only the vision encoder is updated for existing online TTA methods [15, 19, 24, 25] adopted for CLIP.

0.2.1. Experiments on CIFAR-10C, CIFAR-100C, and ImageNet-C

BATCLIP (Ours): For domain-specific test adaptation, we conducted experiments using ViT-B/16 and ViT-B/32 [5] as the vision backbones. For CIFAR-10C, both the vision encoder (f_{vis}) and text encoder (f_{txt}) were updated using the AdamW optimizer with a learning rate of 10^{-3}. Similarly, for CIFAR-100C and ImageNet-C, we employed the Adam optimizer and AdamW optimizer, respectively, with a learning rate of $5{\times}10^{-4}$. The batch size $\mathcal{B}$ used was set to 200 for CIFAR-10C and CIFAR-100C, and 64 for ImageNet-C. Throughout, the prompt template is fixed to "a photo of a <CLS>.".

TENT [24]: We follow all the hyperparameters that TENT provides in their official implementation [2]. To update the vision encoder, we use Adam as the optimizer with a learning rate of 10^{-3} for CIFAR-10C and CIFAR-100C. For ImageNet-C, we update using SGD with a learning rate of $25{\times}10^{-5}$.

RoTTA [25]: For fairness, the batch sizes are set to 200 for the CIFAR datasets and 64 for ImageNet-C. The vision encoder is updated based on the Adam rule with a learning rate of 10^{-3}. The capacity of the memory bank is set to 64, for all the datasets. Following the notations in the paper, α = 0.05, δ = 0.1, ν = 0.001, λ_t and λ_u = 1.0. We implement the details exactly as described in their main paper.

RPL [19]: We use an Adam optimizer with a learning rate of 10^{-3} for CIFAR-10C and CIFAR-100C. For ImageNet-C, the update rule is SGD with a learning rate of $5{\times}10^{-4}$. To compute the generalized cross-entropy loss, q is set to 0.8 for all the datasets.

SAR [15]: The training details/hyperparameters for SAR are the same as RPL [19] for CIFAR-10 and CIFAR-100. For ImageNet-C, the learning rate is set to $25{\times}10^{-5}$ with an SGD update rule. The entropy threshold E_0 is 0.4xln(C), where C is the number of classes. ρ is set to a default of 0.05. The moving average factor is 0.9 for e_m and e_0 is set to 0.2. All parameters are the same as in [15].

TPT [20]: For each test image, 63 augmentations are generated based on random resized crops, yielding a batch of 64 images, in addition to the original test image. The prompt/context vectors are initialized based on "a photo of a <CLS>." and tokenized using pre-trained CLIP weights. The confidence threshold is set to 10% *i.e.,* the marginal entropy over the 10% confident samples is minimized. For all the datasets, we follow their core implementation and optimize the prompt vectors using an AdamW optimizer with a learning rate of $5{\times}10^{-3}$.

VTE [4]: In VTE, an ensemble of different prompt templates is considered based on the idea of [18]. An example of templates includes "a photo of a <CLS>.", "a sketch of a <CLS>.", "a painting of a <CLS>.", etc. The prompt templates are then averaged. On the vision side, similar to TPT [20], a batch of random augmentation is created for a test image with no model updates.

WATT-P* and WATT-S* [16]: The two original variants of WATT [16] were proposed with weight-averaging of adapted weights from multiple prompt templates. Additionally, for each test batch, adaptation was performed over multiple iterations. To fit our online TTA scheme, we reduced the number of iterations to a single step for each prompt template and reset CLIP parameters only after a domain. However, this is still not fully online, as *training* is performed on the test batch using 8 selected prompt templates: "a photo of a <CLS>", "itap of a <CLS>", "a bad photo of the <CLS>", "a origami <CLS>", "a photo of the large <CLS>", "a <CLS> in a video game", "art of the <CLS>", and "a photo of the small <CLS>". As they report performance on CIFAR-10C and CIFAR-100C only, we follow their original implementation details. For experiments using WATT-S*, we set a batch size of 200 (for a fair comparison to other baselines) and a learning rate of 10^{-3} using an Adam optimizer. For WATT-P*, a learning rate of 10^{-4} is used with the same batch size.

Stat*A* [26]: We adopt the original hyperparameter settings of Stat*A* for online TTA in our reported ImageNet experiments and apply them to ImageNet-C. To control the effective number of correlated classes in a test batch, we set γ to 0.1 (low correlation) and -1 (separate, sequential). The default prompt template is "a photo of a <CLS>."

0.2.2. Experiments on Domain Generalization Datasets

We evaluate BATCLIP on standard domain generalization datasets [7]. For a fair comparison, as reported in WATT [16], we use a batch size of 128 for all the online TTA experiments on VLCS, PACS, and Office Home. We use an AdamW optimizer for model updates using BATCLIP with learning rates of $5{\times}10^{-4}$, 10^{-3}, and $5{\times}10^{-3}$ for Office-Home, PACS, and VLCS, respectively. We use the same learning settings for TENT, WATT-S*, and WATT-P*, as in WATT [16]. In addition to the mentioned datasets, we also run experiments on the Terra Incognita dataset and optimize

[1] https://github.com/mariodoebler/test-time-adaptation/tree/main

[2] https://github.com/DequanWang/tent

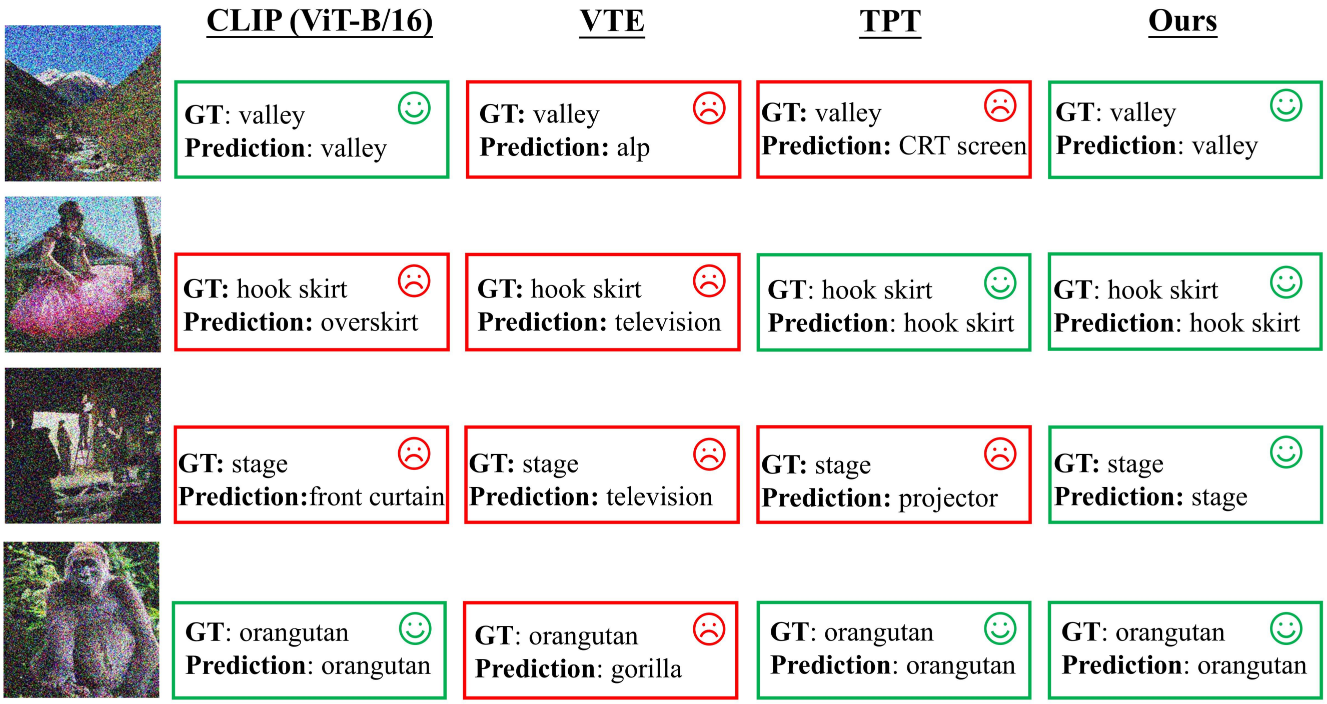

Figure 2. Comparison of classification predictions across various methods (Zero-shot CLIP (ViT-B/16), VTE [4], TPT [20], and Ours) on ImageNet-C samples with *Gaussian* noise. Each row illustrates an example, displaying the ground truth (GT) label alongside the predictions from each method. Correct predictions are highlighted in green, while incorrect ones are marked in red. Our approach demonstrates enhanced robustness and higher accuracy, especially in challenging image corruption conditions.

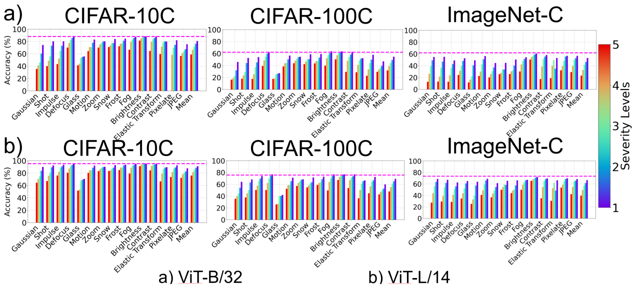

Figure 3. Task-wise mean accuracy (%) of zero-shot CLIP across different corruption severity levels. [Top]: ViT-B/32 backbone. [Bottom]: ViT-L/14 backbone. The **dashed lines** indicate the performance of zero-shot CLIP (w/ respective visual backbones) on the corresponding source datasets.

using an AdamW optimizer with a learning rate of 5×10^{-4} and a batch size of 256.

0.3. Additional Results

In the following subsections, we provide additional results and discussions.

0.3.1. Zero-shot performance analysis of ViT-B/32 and ViT-L/14

In the main paper, we analyze and evaluate the zero-shot performance of ResNet-101 (RN101) [8] and ViT-B/16 [5] and conclude that such CLIP backbones are extremely sensitive, in terms of classification accuracy, to increasing severity levels of image corruption. This could be a major concern in situations involving real-time deployment of CLIP. Here, we present a similar analysis in Figure 3, using ViT-B/32 and ViT-L/14 as backbones. Our analysis, from the main paper, carries forward. To summarise, irrespective of the CLIP visual backbone, the robustness towards image corruption is limited. The classification performance degrades with an increase in the severity of corruption in an image.

0.3.2. Online TTA results using a ViT-B/32 backbone

In the main paper, we had presented the online TTA results on CIFAR-10C, CIFAR-100C, and ImageNet-C, using a ViT-B/16 backbone. In Table 1, we provide results using a ViT-B/32 backbone. Across all the datasets, we see that BATCLIP achieves the best or comparable performance against all the baseline approaches.

	Method	Venue	*Gaussian*	*Shot*	*Impulse*	*Defocus*	*Glass*	*Motion*	*Zoom*	*Snow*	*Frost*	*Fog*	*Brightness*	*Contrast*	*Elastic*	*Pixelate*	*JPEG*	Mean
CIFAR-10C	ZS	ICLR'21	35.47	39.94	43.23	69.95	41.43	64.50	70.13	70.85	72.33	66.66	81.37	64.57	59.69	48.28	56.62	59.00
	TENT	ICLR'21	20.09	23.45	34.47	69.85	23.01	39.79	60.35	76.83	77.49	76.07	**88.88**	**81.38**	65.35	57.01	51.19	56.35
	RoTTA	CVPR'23	36.55	40.91	43.99	70.03	42.45	64.52	70.08	71.23	72.68	67.31	81.92	64.99	60.33	49.40	57.11	59.57
	RPL	arXiv	15.89	19.08	34.04	**77.84**	18.72	41.22	62.39	78.17	78.86	**76.31**	88.83	81.15	68.98	54.19	51.91	56.51
	SAR	ICML'22	50.28	54.12	49.65	73.08	51.98	71.17	74.65	73.73	75.22	70.99	84.25	72.08	63.93	51.57	60.32	65.13
	TPT	NeurIPS'22	43.11	46.53	48.29	71.31	47.80	66.89	71.96	74.00	76.00	68.81	84.12	66.35	63.86	51.86	58.01	62.59
	VTE	ECCV-W'24	47.59	50.18	**53.15**	71.39	53.86	67.92	72.90	76.37	76.30	70.78	83.27	61.07	69.00	58.57	61.14	64.90
	WATT-P*	NeurIPS'24	43.64	47.1	45.97	74.98	48.04	70.42	74.74	76.1	76.86	71.85	85.15	70.36	64.6	56.0	60.37	64.41
	WATT-S*	NeurIPS'24	**56.38**	**58.08**	52.48	78.07	**58.29**	**76.42**	**79.16**	**78.9**	**79.71**	76.76	87.19	76.49	**70.35**	**64.11**	**65.05**	**70.49**
	Ours		52.39	55.99	52.54	76.79	54.04	74.90	75.79	77.67	79.10	75.31	86.33	77.34	67.41	57.06	61.29	68.26
CIFAR-100C	ZS	ICLR'21	16.23	17.83	17.57	39.07	17.63	38.55	43.81	42.32	43.46	39.71	50.32	29.34	28.74	22.85	29.42	31.79
	TENT	ICLR'21	5.53	7.64	6.85	**49.60**	4.47	48.45	52.35	49.77	26.77	37.50	63.05	**50.53**	13.89	27.00	30.80	31.61
	RoTTA	CVPR'23	16.63	18.25	17.78	38.62	17.76	38.38	43.52	42.39	43.41	39.37	50.60	28.85	28.89	23.50	29.65	31.84
	RPL	arXiv	4.50	5.80	9.61	50.26	4.43	**48.88**	**52.61**	**50.27**	22.36	25.34	**63.36**	50.31	9.10	18.65	**34.53**	30.00
	SAR	ICML'22	**24.63**	**27.14**	21.25	44.57	22.98	43.95	48.40	48.01	**47.76**	44.85	57.76	42.11	32.69	**28.02**	33.08	**37.81**
	TPT	NeurIPS'22	16.08	17.65	17.54	39.21	19.47	38.91	44.01	43.45	44.46	40.15	50.93	27.77	30.91	23.36	29.55	32.23
	VTE	ECCV-W'24	16.84	18.33	18.94	39.63	22.88	39.13	43.80	44.56	44.88	39.21	49.37	28.37	34.13	26.87	30.12	33.14
	WATT-P*	NeurIPS'24	15.55	17.02	16.16	40.25	16.16	38.74	43.6	42.49	43.51	39.39	51.17	31.85	28.35	23.94	29.74	31.86
	WATT-S*	NeurIPS'24	16.22	17.72	16.85	41.54	17.04	39.66	44.55	43.33	44.26	40.26	52.13	33.13	29.34	24.65	30.39	32.73
	Ours		21.35	24.71	**22.32**	46.26	**23.07**	44.64	50.12	47.23	46.88	**44.92**	58.55	38.52	**34.56**	27.73	33.19	37.60
ImageNet-C	ZS	ICLR'21	12.88	13.04	12.90	24.42	11.86	22.72	20.20	25.70	25.84	30.28	50.54	17.32	18.96	32.20	29.12	23.20
	TENT	ICLR'21	9.18	8.50	10.42	**26.02**	15.72	26.06	21.64	27.12	26.18	31.60	50.58	22.28	20.12	34.06	31.30	24.05
	RoTTA	CVPR'23	13.10	13.30	13.02	24.48	11.96	22.86	20.28	26.06	26.06	30.24	50.46	17.34	19.16	32.44	29.18	23.33
	RPL	arXiv	11.68	10.98	12.10	25.68	13.24	23.98	20.84	26.32	26.12	30.86	50.62	19.30	19.48	33.14	29.92	23.62
	SAR	ICML'22	**19.82**	**20.36**	**20.92**	25.78	20.40	28.34	23.10	28.12	**28.38**	34.74	51.10	**24.60**	24.38	**36.54**	34.40	**28.07**
	TPT	NeurIPS'22	12.04	12.64	12.52	25.38	12.28	22.68	20.78	26.36	26.64	30.78	51.02	16.50	19.90	33.62	30.62	23.58
	VTE	ECCV-W'24	11.96	12.32	13.44	25.06	11.70	22.58	22.40	27.38	27.02	32.28	**51.52**	16.84	19.94	34.80	32.82	24.14
	Stat*A* (γ=0.1)	CVPR'25	11.57	12.28	11.74	21.91	10.91	20.43	18.87	24.04	25.0	29.12	49.21	16.02	18.87	29.15	26.72	21.73
	Stat*A* (γ=-1)	CVPR'25	12.52	13.10	12.74	22.43	11.39	21.17	19.61	24.37	25.60	29.44	49.89	17.71	19.21	29.87	27.52	22.44
	Ours		16.84	18.20	16.10	25.04	**20.90**	**28.90**	**25.24**	**29.42**	27.18	**36.02**	50.18	17.66	**27.68**	36.20	**35.42**	27.39

Table 1. Mean accuracy (%) on CIFAR-10C, CIFAR-100C, and ImageNet-C - TTA mean accuracy of the 15 corruptions (tasks) at a severity level of 5, using ViT-B/32.

0.3.3. Detailed results from the loss ablation study

In the main paper, we provide ablation of loss components and their combinations *i.e.,* the mean accuracy across all the tasks for ViT-B/16 on the benchmark corruption datasets. Here, we provide additional task-wise accuracy in Table 2. The addition of loss components $\mathcal{L}_{pm}$ and $\mathcal{L}_{sp}$ help CLIP in adapting its feature space to a specific domain/corruption.

0.3.4. Effect of different prompt templates

In Table , we show results with "relevant" prompt templates to show the independence of such prompt selection, at test-time. As seen, the performance gain over zero-shot ViT-B/32 is fairly large for all the prompt templates. Though TPT [20] fine-tunes a pre-trained prompt on each image, and VTE [4] uses an ensemble of prompts, our method is agnostic to the prompt template being used, making it favorable for real-time usage.

In all of our prior experiments, we use a generic prompt template "a photo of a <CLS>." for all of the datasets and methods. Here, we replace this with "relevant" prompt templates to show the independence of such a prompt selection, at test-time, and report the results in Table 5. As seen, the performance gain over zero-shot ViT-B/32 is fairly large for all the prompt templates. Though TPT [20] fine-tunes a pre-trained prompt on each test image, and VTE [4] uses an ensemble of prompts, our method is agnostic to the prompt template being used, making it favorable for real-time deployment.

0.3.5. Impact of *bimodal* adaptation

To show the effectiveness of *bimodal* adaptation, we ablate $\mathcal{L}_{pm}$, which is responsible for updating the text encoder f_{txt}. We report the results in Table 6. We see a drop in accuracy when f_{txt} is "frozen" i.e., when $\mathcal{L}_{pm}$ isn't used, necessitating the need for *bimodal* adaptation of CLIP encoders.

0.3.6. BATCLIP for other vision-language models

We employ SigLIP [27], a recent pre-trained vision-language model with 877 million parameters, for our online TTA experiments on ImageNet-C. We use a batch size of 8 and a learning rate of 5×10^{-5} with the AdamW optimizer. Despite the small batch size, which could impact prototypes, BATCLIP achieves better performance, as shown in Table 8.

0.3.7. Results on distribution shifts caused by lighting conditions, camera types, or object scales

To evaluate BATCLIP across a wide spectrum of shifts, we extend our setup to datasets exhibiting variations due to lighting conditions, camera types, and object scales. We

	Method	*Gaussian*	*Shot*	*Impulse*	*Defocus*	*Glass*	*Motion*	*Zoom*	*Snow*	*Frost*	*Fog*	*Brightness*	*Contrast*	*Elastic*	*Pixelate*	*JPEG*	Mean
CIFAR-10C	ViT-B/16																
	$\mathcal{L}_{ent}$	14.62	17.29	49.25	**81.06**	20.23	74.27	81.10	**84.25**	81.93	**80.93**	**91.86**	78.92	53.09	51.18	49.79	60.65
	$\mathcal{L}_{pm}$	41.82	45.66	52.88	70.13	39.29	66.61	71.27	71.49	74.71	69.20	80.98	72.89	55.58	55.55	50.44	61.23
	$\mathcal{L}_{sp}$	62.47	65.43	63.41	79.96	52.73	80.02	81.38	82.35	83.44	80.46	88.85	81.22	67.77	60.52	67.74	73.16
	$\mathcal{L}_{ent}$+$\mathcal{L}_{pm}$	16.58	19.89	42.69	79.45	23.41	77.03	80.95	81.74	78.45	80.66	90.52	**82.55**	62.56	**64.35**	58.16	62.60
	$\mathcal{L}_{ent}$ + 0.1 *($\mathcal{L}_{pm}$ + $\mathcal{L}_{sp}$)	44.92	51.41	63.30	80.65	50.79	79.83	83.13	83.59	83.89	81.91	89.56	83.16	67.78	66.69	67.87	71.90
	$\mathcal{L}_{ent}$ + 0.5 *($\mathcal{L}_{pm}$ + $\mathcal{L}_{sp}$)	58.81	63.85	65.99	80.26	53.90	80.30	82.30	83.08	84.04	81.66	89.19	82.67	68.29	62.70	68.52	73.70
	$\mathcal{L}_{ent}$+$\mathcal{L}_{pm}$+$\mathcal{L}_{sp}$	**61.13**	**64.09**	**65.76**	80.51	**54.96**	**80.65**	**81.94**	83.04	**84.19**	80.84	88.95	82.15	**69.16**	62.68	**67.64**	**73.85**
CIFAR-100C	ViT-B/16																
	$\mathcal{L}_{ent}$	7.71	10.05	11.52	49.42	12.49	**49.36**	53.79	**54.11**	50.76	**49.92**	**64.32**	**47.07**	33.40	**38.63**	**39.95**	38.17
	$\mathcal{L}_{pm}$	19.88	24.06	21.26	45.57	22.66	43.66	49.37	46.05	45.96	43.69	57.76	37.33	33.66	25.85	32.65	36.63
	$\mathcal{L}_{sp}$	24.69	27.28	33.62	49.08	25.29	47.84	53.86	51.70	50.93	46.93	62.57	44.76	33.88	31.89	36.05	41.36
	$\mathcal{L}_{ent}$+$\mathcal{L}_{pm}$	12.26	12.62	13.14	48.90	26.22	48.99	53.10	53.10	**52.43**	49.44	63.36	46.78	33.27	37.77	38.36	39.32
	$\mathcal{L}_{ent}$ + 0.1 *($\mathcal{L}_{pm}$ + $\mathcal{L}_{sp}$)	24.99	27.10	32.95	49.92	25.91	48.42	54.43	52.87	51.57	47.70	63.68	45.25	39.74	31.58	37.16	41.88
	$\mathcal{L}_{ent}$ + 0.5 *($\mathcal{L}_{pm}$ + $\mathcal{L}_{sp}$)	25.24	27.59	33.41	50.05	25.73	48.55	54.44	52.85	51.80	47.81	63.75	45.08	34.63	31.67	37.17	41.98
	$\mathcal{L}_{ent}$+$\mathcal{L}_{pm}$+$\mathcal{L}_{sp}$	**24.91**	**27.73**	**33.66**	**50.11**	**26.27**	48.49	**54.85**	52.35	51.62	48.38	63.27	45.21	**34.74**	32.38	37.31	**42.09**
ImageNet-C	ViT-B/16																
	$\mathcal{L}_{ent}$	0.90	1.06	1.16	**29.12**	13.02	**32.14**	27.34	35.32	11.14	40.92	**56.90**	23.78	7.78	39.62	40.22	24.03
	$\mathcal{L}_{pm}$	10.20	11.40	10.74	19.56	15.18	20.06	19.28	27.66	29.72	34.10	53.50	22.66	13.80	24.38	30.26	22.83
	$\mathcal{L}_{sp}$	19.32	20.98	19.26	25.90	21.22	30.06	28.56	35.22	31.34	40.36	55.20	25.64	23.68	36.90	37.18	30.05
	$\mathcal{L}_{ent}$+$\mathcal{L}_{pm}$	0.90	1.16	1.30	28.90	17.04	31.56	26.24	36.26	12.22	**42.12**	57.92	**30.34**	10.36	**40.66**	**41.20**	25.21
	$\mathcal{L}_{ent}$ + 0.1 *($\mathcal{L}_{pm}$ + $\mathcal{L}_{sp}$)	19.48	21.14	19.16	26.80	20.70	29.80	29.32	35.92	30.68	41.04	55.80	25.66	22.50	37.68	37.92	30.25
	$\mathcal{L}_{ent}$ + 0.5 *($\mathcal{L}_{pm}$ + $\mathcal{L}_{sp}$)	19.56	21.38	19.16	26.96	21.26	30.06	29.24	35.92	31.26	41.34	56.32	25.74	22.58	37.94	37.92	30.44
	$\mathcal{L}_{ent}$+$\mathcal{L}_{pm}$+$\mathcal{L}_{sp}$	**19.32**	**21.38**	**19.60**	26.58	**21.94**	30.88	**29.02**	**36.48**	32.00	40.98	56.72	26.14	**23.74**	37.68	38.34	**30.72**

Table 2. Task-wise loss ablation results (accuracy) on CIFAR-10C, CIFAR-100C, and ImageNet-C.

Method	*Gaussian*	*Shot*	*Impulse*	*Defocus*	*Glass*	*Motion*	*Zoom*	*Snow*	*Frost*	*Fog*	*Brightness*	*Contrast*	*Elastic*	*Pixelate*	*JPEG*	Zero-Shot
ViT-B/16																90.1
Ours	84.51	84.29	88.69	88.48	86.44	86.46	87.04	91.38	91.01	90.22	90.87	88.12	88.13	74.74	87.48	87.19 (mean)
ViT-B/32																88.3
Ours	67.41	68.28	84.23	80.50	75.37	79.75	78.55	87.67	86.36	85.83	90.04	80.89	81.56	82.74	82.36	80.77 (mean)

Table 3. Zero-shot performance on CIFAR10 (source) after adaptation of `BATCLIP` on a task.

Method	*Gaussian*	*Shot*	*Impulse*	*Defocus*	*Glass*	*Motion*	*Zoom*	*Snow*	*Frost*	*Fog*	*Brightness*	*Contrast*	*Elastic*	*Pixelate*	*JPEG*	Zero-Shot
ViT-B/16																66.6
Ours	67.09	67.12	67.08	70.31	63.70	66.83	70.12	70.10	68.19	69.55	71.05	66.49	65.13	59.98	67.60	67.36 (mean)
ViT-B/32																62.3
Ours	45.99	46.90	60.86	63.62	57.73	59.59	61.66	66.21	61.50	63.61	66.39	57.05	62.13	62.82	65.37	60.09 (mean)

Table 4. Zero-shot performance on CIFAR100 (source) after adaptation of `BATCLIP` on a task.

conduct experiments on ImageNet-ES [1], which introduces significant variations in lighting and camera sensor settings (e.g., ISO, shutter speed). While more details can be found in the original work, but ImageNet-ES introduces wide variations in lighting conditions and camera sensor factors (ISO, shutter speed, etc.). We report the results in Table 7 using a ViT-B/16 backbone. To be noted that, “Auto-Exposure” has 5 tasks for each condition, while “Manual” has 27. On average, `BATCLIP` outperforms all the reported baselines.

0.3.8. Post-adaptation results on source test sets

Thanks to the natural language supervision and also due to the pre-training on large amounts of (image, text) pairs, CLIP has shown strong generalization capabilities. However, for an efficient adaptation to a downstream task, fine-

Prompt Template	CIFAR-10C	CIFAR-100C	ImageNet-C
"a low contrast photo of a <CLS>."	68.53 (**+7.81**)	37.09 (**+4.97**)	27.31 (**+3.71**)
"a blurry photo of a <CLS>."	68.84 (**+10.96**)	36.80 (**+5.33**)	26.92 (**+3.52**)
"a photo of a big <CLS>."	67.49 (**+10.10**)	35.79 (**+4.87**)	25.64 (**+3.29**)

Table 5. Prompt template selection. **+** denotes the accuracy gain over zero-shot ViT-B/32.

Dataset	f_{vis} update	f_{txt} update	Ours
CIFAR-10C	✔	✗	72.58
	✔	✔	**73.85**
CIFAR-100C	✔	✗	41.00
	✔	✔	**42.09**
ImageNet-C	✔	✗	29.88
	✔	✔	**30.72**

Table 6. Ablation on $\mathcal{L}_{pm}$ to demonstrate the need of *bimodal* adaptation of CLIP encoders - using a ViT-B/16 backbone.

Camera Sensor	Condition	ZS	TENT	SAR	Ours
Auto-Exposure	Light on	50.90	51.62	52.60	**53.82**
	Light off	46.96	47.44	47.88	**49.32**
Manual	Light on	60.44	60.64	60.30	**61.26**
	Light off	60.76	60.99	59.62	**61.61**

Table 7. Online TTA experiments on ImageNet-ES [1]. Mean accuracy (in %).

SigLIP	Clean (ImageNet)	Source	TENT	SAR	BATCLIP
ImageNet-C	82.00	35.44	37.58	39.62	**40.10**

Table 8. Online TTA experiments on ImageNet-C with SigLIP [27].

tuning the full model is infeasible due to large model updates. The primary reason is the loss of useful pre-trained knowledge of CLIP, which could eventually lead to overfitting to a downstream task. However, for attention-based models, tuned for multimodal tasks, [28] show that tuning the *LayerNorm* parameters leads to strong results. Inspired by [28], in our *bimodal* test-adaptation scheme, we update the *LayerNorm* parameters of both CLIP encoders, to a specific corruption task, which makes it parametric-efficient. We perform a single-domain TTA or adapt CLIP, at test-time, to a single domain only and then reset the parameters. Now, with continual adaptation to a certain corruption task, it gets difficult to preserve CLIP's pre-trained knowledge since the normalization parameters begin to overfit to this domain. Then, a natural question arises -

Given that CLIP has been adapted to a specific corruption task, will the zero-shot generalization still hold back on its source test set?

In this crucial experiment, we challenge our BATCLIP, and evaluate its zero-shot generalization performance back on the source test set, to check the preservation of pre-trained CLIP knowledge. After the adaptation of CLIP on each corruption task, we report the adapted model's zero-shot performance on its corresponding source test set. We report results for CIFAR-10C and CIFAR-100C in Tables 3 and 4, using ViT-B/16 and ViT-B/32 backbones. For all of the results, we use the prompt template "a photo of a <CLS>.". As an example, for CIFAR-10C, upon adaptation of CLIP to *Gaussian noise* following our approach, we report the adapted model's zero-shot accuracy on its source test set - CIFAR10 test set.

From Table 3, we observe that, on average, there is a 2.91% drop in accuracy compared to a zero-shot evaluation using pre-trained CLIP ViT-B/16. Similarly, for ViT-B/32, we see a drop of about 7.53% in mean accuracy. In Table 4, for CIFAR-100C using a ViT-B/16 backbone, we see an improvement of 0.76% in mean accuracy.

On the whole, we conclude that since the adaptation for a task happens over multiple test batches, the zero-shot performance back on the source data largely depends on the distribution of the image corruption. Overall, ViT-B/16 visual backbones preserve larger amounts of CLIP pre-trained knowledge. This proves the effectiveness of our method BATCLIP, on average.

0.3.9. t-SNE visualizations of CIFAR-10C and CIFAR-100C

In this section, we provide illustrations of task-wise t-SNE [22] plots for CIFAR-10C and CIFAR-100C and compare them against zero-shot ViT-B/16. The results are in Figures 4, 5, 6, and 7. Across all corruptions/tasks, BATCLIP learns strong discriminative visual features with a strong image-text alignment and class-level separation. ImageNet-C has 1000 classes, so, we do not provide t-SNE plots to avoid complications. However, the analysis and results carry forward.

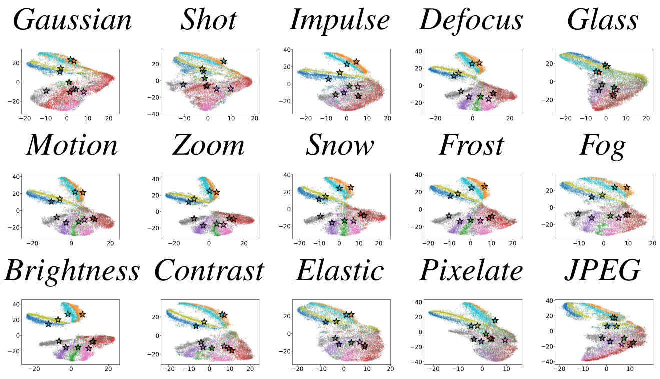

Figure 4. BATCLIP (w/ ViT-B/16): The t-SNE plots show visual (○) and text (★) features for CIFAR-10C.

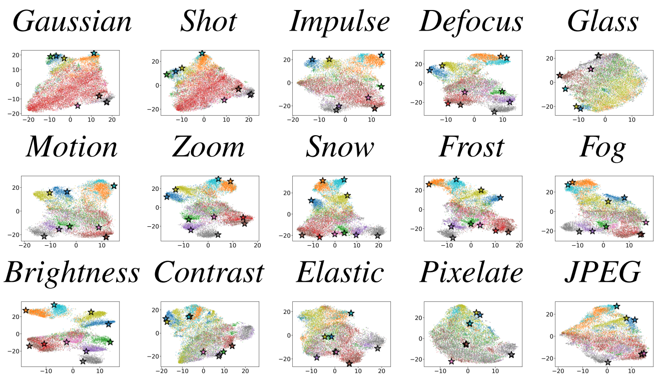

Figure 5. Zero-shot ViT-B/16: The t-SNE plots show visual (○) and text (★) features for CIFAR-10C.

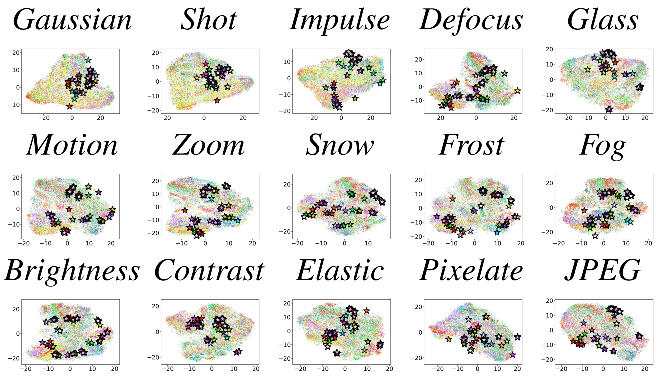

Figure 6. `BATCLIP` (w/ ViT-B/16): The t-SNE plots show visual (○) and text (★) features for CIFAR-100C.

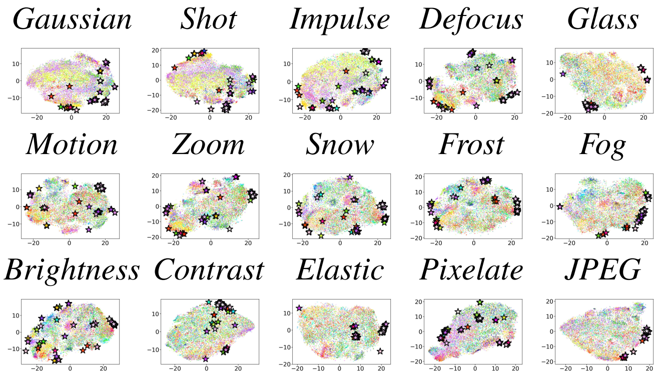

Figure 7. Zero-shot ViT-B/16: The t-SNE plots show visual (○) and text (★) features for CIFAR-100C.

References

[1] Eunsu Baek, Keondo Park, Jiyoon Kim, and Hyung-Sin Kim. Unexplored faces of robustness and out-of-distribution: Covariate shifts in environment and sensor domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22294–22303, 2024. 5, 6

[2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018. 1

[3] 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*, pages 248–255. IEEE, 2009. 1

[4] Mario Döbler, Robert A Marsden, Tobias Raichle, and Bin Yang. A lost opportunity for vision-language models: A comparative study of online test-time adaptation for vision-language models. *ECCV Workshops*, 2024. 2, 3, 4

[5] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[6] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE international conference on computer vision*, pages 1657–1664, 2013. 1

[7] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020. 1, 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3

[9] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 1, 2

[10] Dan Hendrycks and Kevin Gimpel. Early methods for detecting adversarial images. *arXiv preprint arXiv:1608.00530*, 2016. 1

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[12] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 1

[13] Chang Liu, Yinpeng Dong, Wenzhao Xiang, Xiao Yang, Hang Su, Jun Zhu, Yuefeng Chen, Yuan He, Hui Xue, and Shibao Zheng. A comprehensive study on robustness of image classification models: Benchmarking and rethinking. *International Journal of Computer Vision*, pages 1–23, 2024. 1

[14] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In *International Conference on Learning Representations*, 2017. 1

[15] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Machine Learning*, pages 16888–16905. PMLR, 2022. 1, 2

[16] David Osowiechi, Mehrdad Noori, Gustavo Adolfo Vargas Hakim, Moslem Yazdanpanah, Ali Bahri, Milad Cheraghalikhani, Sahar Dastani, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. Watt: Weight average test-time

adaption of clip. *arXiv preprint arXiv:2406.13875*, 2024. 1, 2

[17] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016. 1

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2

[19] Evgenia Rusak, Steffen Schneider, George Pachitariu, Luisa Eck, Peter Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. If your data distribution shifts, use self-learning. *arXiv preprint arXiv:2104.12928*, 2021. 1, 2

[20] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022. 2, 3, 4

[21] Adarsh Subbaswamy, Roy Adams, and Suchi Saria. Evaluating model robustness and stability to dataset shift. In *International Conference on Artificial Intelligence and Statistics*, pages 2611–2619. PMLR, 2021. 1

[22] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (11), 2008. 6

[23] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 1

[24] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 1, 2

[25] 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*, pages 15922–15932, 2023. 1, 2

[26] Maxime Zanella, Clément Fuchs, Christophe De Vleeschouwer, and Ismail Ben Ayed. Realistic test-time adaptation of vision-language models. *arXiv preprint arXiv:2501.03729*, 2025. 2

[27] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 4, 6

[28] Bingchen Zhao, Haoqin Tu, Chen Wei, Jieru Mei, and Cihang Xie. Tuning layernorm in attention: Towards efficient multi-modal llm finetuning. *arXiv preprint arXiv:2312.11420*, 2023. 6