SEMANTIC ANCHOR TRANSPORT: ROBUST TEST-TIME ADAPTATION FOR VISION-LANGUAGE MODELS

Shambhavi Mishra* **Julio Silva-Rodríguez** **Ismail Ben Ayed**
Marco Pedersoli Jose Dolz

LIVIA, ÉTS Montréal, Canada
International Laboratory on Learning Systems (ILLS),
McGILL - ETS - MILA - CNRS - Université Paris-Saclay - CentraleSupélec, Canada

* shambhavi.mishra.1@etsmtl.net

ABSTRACT

Large pre-trained vision-language models (VLMs) like CLIP exhibit strong zero-shot performance but struggle under distributional shifts. We propose Semantic Anchor Transport (SAT), a method that generates pseudo-labels for test samples by aligning visual embeddings with reliable text-based semantic anchors using Optimal Transport for batch-wise label assignment. These pseudo-labels enable efficient test-time adaptation through principled cross-modal alignment. We further incorporate multi-template distillation to leverage diverse textual clues, replicating multi-view contrastive learning without added computational cost. Extensive experiments demonstrate consistent performance gains over state-of-the-art methods across multiple benchmarks while maintaining computational efficiency.

1 INTRODUCTION

Large pre-trained vision-language models (VLMs) like CLIP Radford et al. (2021) exhibit strong zero-shot performance but struggle under distributional shifts. Recent test-time adaptation (TTA) methods generate pseudo-labels to supervise model updates Osowiechi et al. (2024); Maharana et al. (2025); Hakim et al. (2025), typically minimizing cross-entropy between predictions and pseudo-labels derived from those same predictions. However, this creates a feedback loop: when a prediction is wrong, the pseudo-label reinforces the error, a problem we term *error accumulation*.

We demonstrate this empirically in Figure 1. Starting from samples that zero-shot CLIP misclassifies on CIFAR-10C, we track cosine similarity to both the predicted (wrong) class and the ground-truth class across adaptation steps. TENT Wang et al. (2021), WATT Osowiechi et al. (2024), and BATCLIP Maharana et al. (2025) all *increase* similarity to the wrong class while *decreasing* similarity to the correct class, confirming they reinforce errors.

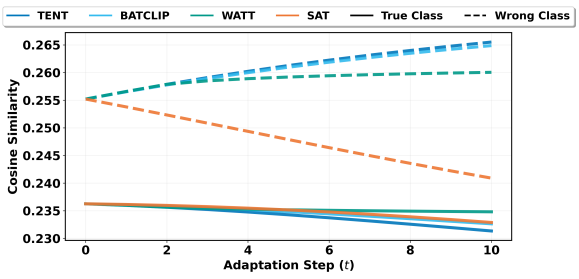

Figure 1: Tracking samples misclassified by zero-shot CLIP on CIFAR-10C across adaptation steps. We plot cosine similarity to the predicted (incorrect) class and the true class. TENT, WATT, and BATCLIP increase similarity to the incorrect class (dashed lines). SAT is the only method where similarity to the incorrect class decreases.

We address this by reformulating TTA as a cross-modal alignment problem Yuan et al. (2023), where test images are aligned to fixed semantic anchors provided by text prototypes. We leverage batch structure to find globally optimal assignments between visual embeddings and text-based class prototypes. We formulate this as an Optimal Transport (OT) problem Cuturi (2013), which naturally handles multi-modal distributions Lee et al. (2019) and prevents degenerate solutions through uniform marginal constraints.

Contributions: (1) We propose Semantic Anchor Transport (SAT), a novel framework that casts pseudo-labeling in CLIP TTA as batch-wise Optimal Transport assignment, leveraging text-based semantic anchors as robust cluster centroids. (2) We employ the Sinkhorn algorithm Cuturi (2013) for efficient label assignment that handles multi-modal distributions. (3) We introduce multi-template knowledge distillation that leverages diverse text prompts to guide adaptation without significant computational overhead. (4) Experiments on visual corruptions and domain shift benchmarks demonstrate SAT's superiority over state-of-the-art methods while maintaining computational efficiency.

2 METHOD

Problem Setting. We address adapting a pre-trained VLM at test time. Given a model trained on source domain $\mathcal{D}_S$, our goal is to adapt it online to target domain $\mathcal{D}_T$, where only unlabeled test data $\{\boldsymbol{x}_i\}_{i=1}^{B_T}$ arrives as a stream of batches. CLIP Radford et al. (2021) consists of a visual encoder $\boldsymbol{\theta}$ producing embeddings $\mathbf{z} = \boldsymbol{\theta}(\boldsymbol{x}) \in \mathbb{R}^d$ and a text encoder $\boldsymbol{\phi}$. For K classes with M templates, we generate text prototypes $\mathbf{T} = \{\mathbf{t}_k\}_{k=1}^K$ where $\mathbf{t}_k = \frac{1}{M}\sum_{m=1}^M \boldsymbol{\phi}(T_{mk})$. Zero-shot prediction follows:

$$p(y{=}k|\boldsymbol{x}_i) = \frac{\exp(\mathbf{z}_i^\top \mathbf{t}_k/\tau)}{\sum_{j=1}^K \exp(\mathbf{z}_i^\top \mathbf{t}_j/\tau)} \tag{1}$$

A straightforward TTA approach minimizes Shannon entropy Wang et al. (2021):

$$\mathcal{L}(p) = -\frac{1}{B_T}\sum_{i=1}^{B_T}\sum_{k=1}^{K} p(y=k|\boldsymbol{x}_i)\log p(y=k|\boldsymbol{x}_i). \tag{2}$$

However, this risks degenerate solutions, as Eq. equation 2 can be trivially minimized by assigning all samples to a single class. TTA literature addresses this by introducing pseudo-labels $q(y|\boldsymbol{x}_i)$:

$$\mathcal{L}(p,q) = -\frac{1}{B_T}\sum_{i=1}^{B_T}\sum_{k=1}^{K} q(y=k|\boldsymbol{x}_i)\log p(y=k|\boldsymbol{x}_i). \tag{3}$$

The critical challenge is constructing high-quality pseudo-labels $q_{ik} = q(y=k|\boldsymbol{x}_i)$ without ground truth. Prior methods Wang et al. (2021); Osowiechi et al. (2024); Hakim et al. (2025); Shu et al. (2022); Maharana et al. (2025) derive $\mathbf{Q} = [q_{ik}]$ from per-sample predictions, creating a self-referential loop. As shown in Figure 1, this leads to *error accumulation*: incorrect predictions generate faulty supervision that reinforces mistakes.

Batch-Aware Cross-Modal Alignment We reframe pseudo-label generation as finding optimal alignment between the batch's visual features $\mathbf{Z} \in \mathbb{R}^{d\times B_T}$ and text-based semantic anchors $\mathbf{T} \in \mathbb{R}^{d\times K}$. Expressing Eq. equation 3 in matrix form (where $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_K]$ and $\mathbf{Q}$ has columns $\mathbf{q}_i$), and noting the logarithmic term doesn't depend on $\mathbf{q}$, yields:

$$\max_{\mathbf{Q}\in\mathcal{Q}} \operatorname{tr}(\mathbf{Q}^\top \mathbf{T}^\top \mathbf{Z}). \tag{4}$$

Following Cuturi (2013), we constrain $\mathbf{Q}$ to lie in the transportation polytope:

$$\mathcal{Q} = \{\mathbf{Q} \in \mathbb{R}_+^{K\times B_T} \mid \mathbf{Q}\mathbf{1}_{B_T} = \tfrac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top\mathbf{1}_K = \tfrac{1}{B_T}\mathbf{1}_{B_T}\}, \tag{5}$$

where $\mathbf{1}_K$ and $\mathbf{1}_{B_T}$ denote vectors of ones.

Why OT Breaks Error Accumulation. The constraints in Eq. equation 5 enforce uniform marginals, requiring each prototype to be selected at least $\frac{B_T}{K}$ times on average. This *breaks the direct dependency* between prediction p and posterior q, as assignments consider the whole batch distribution. Even when individual samples are confidently wrong, global optimization must distribute assignments across all classes, providing corrective signal.

Efficient Solution via Sinkhorn. Direct optimization of Eq. equation 4 is slow. We apply entropic regularization, yielding:

$$\max_{\mathbf{Q}\in\mathcal{Q}} \operatorname{tr}(\mathbf{Q}^\top \mathbf{T}^\top \mathbf{Z}) + \varepsilon \mathrm{Ent}(\mathbf{Q}), \tag{6}$$

where $\text{Ent}(\mathbf{Q}) = -\sum_{ij} Q_{ij} \log Q_{ij}$ and ε controls regularization. The soft assignment $\mathbf{Q}^*$ is efficiently computed via Sinkhorn Cuturi (2013):

$$\mathbf{Q}^* = \text{Diag}(\mathbf{u}^{(t)}) \exp\left(\frac{\mathbf{T}^\top \mathbf{Z}}{\lambda}\right) \text{Diag}(\mathbf{v}^{(t)}), \tag{7}$$

with renormalization vectors $\mathbf{u} \in \mathbb{R}^K$ and $\mathbf{v} \in \mathbb{R}^{B_T}$.

Multi-Template Distillation. Following Osowiechi et al. (2024), we leverage each of M templates as a distinct semantic view $\mathbf{T}_m = [\mathbf{t}_{1m}, ..., \mathbf{t}_{Km}]$, disentangling two goals: ① **Assignment (Specificity):** We compute a separate transport plan $\mathbf{Q}^*_m$ per template via Eq. equation 7, providing unambiguous matching for each context. ② **Generalization (Consistency):** Predictions $p(y = k|\boldsymbol{x}_i)$ always use the averaged prototypes $\mathbf{T} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{T}_m$, ensuring stability across views and preventing template-specific overfitting. We iterate through M views, computing $\ell(\mathbf{p}_i, \mathbf{q}^*_{im}) = -{\mathbf{q}^*_{im}}^\top \log \mathbf{p}_i$ where $\mathbf{p}_i$ uses averaged $\mathbf{T}$. This mimics multi-view contrastive learning with text embeddings computed *once offline*.

Algorithm. Following Wang et al. (2021); Osowiechi et al. (2024); Hakim et al. (2025), we update only LayerNorm parameters. Per batch, we perform M updates: *(i)* compute $\mathbf{Z}$ and find $\mathbf{Q}^*_m$ by solving Eq. equation 6 via Sinkhorn (Eq. equation 7 with template-specific $\mathbf{T}_m$); *(ii)* update parameters via SGD on Eq. equation 3.

3 Experiments

Datasets and Setup. We evaluate SAT across four visual corruption benchmarks (CIFAR-10C, CIFAR-100C, Tiny-ImageNet-C, ImageNet-C) and four domain shift benchmarks (PACS, Office-Home, VLCS, VisDA-C). We use a CLIP ViT-B/32 backbone with a batch size of 128.

Table 1: Mean Accuracy (%) comparison across all benchmarks. Best results are in **bold**, second best are underlined.

Method	C10-C	C100-C	TIN-C	IN-C	PACS	O-Home	VLCS	VisDA	Mean
CLIP	59.22	29.43	22.15	20.47	93.65	77.53	80.16	84.44	58.38
TENT	67.56	35.19	23.22	21.10	93.81	77.68	80.27	84.77	60.45
TPT	56.80	30.46	25.32	20.01	93.23	77.20	74.57	81.46	57.38
WATT	73.82	45.57	27.87	22.33	94.80	78.83	**81.14**	85.03	63.68
BATCLIP	68.62	33.43	29.32	24.95	94.52	78.90	80.78	82.79	61.66
SAT (Ours)	**77.06**	**47.33**	**34.91**	**26.01**	**95.94**	**80.15**	78.33	**88.09**	**66.10**

Main Results. As summarized in Table 1; detailed in Appendix B.3, SAT establishes a new state-of-the-art across all eight evaluated benchmarks. On common visual corruptions (CIFAR-C, Tiny-ImageNet-C), SAT outperforms the second-best method, WATT, by an average of 3.2%. The gap is particularly significant on Tiny-ImageNet-C (+5.6%) and ImageNet-C (+5.5%), suggesting that our batch-aware Optimal Transport (OT) assignment is more robust as the number of classes increases. In domain generalization benchmarks, SAT demonstrates superior stability. While methods like BATCLIP show significant performance drops on VisDA-C compared to zero-shot models, SAT provides a consistent improvement. This underscores a critical finding: methods relying too heavily on their own initial pseudo-labels often collapse under large domain gaps, whereas SAT's use of fixed semantic anchors and global assignment prevents such divergence.

Impact of each component. Refining LayerNorm parameters via **Average Template Adaptation** further aligns the model to the target domain. Finally, our full method with **Multi-Template Distillation** provides the largest gains by leveraging diverse textual semantic views to supervise the adaptation.

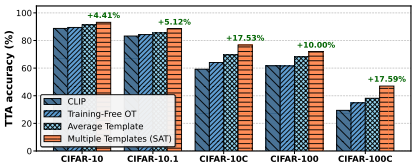

Figure 2: **Ablation analysis**. Green bars denote gains over zero-shot CLIP, highlighting the necessity of each SAT element.

Do observations hold across backbones? Table 2 reports the performance across different datasets when larger CLIP pre-trained models are employed.

Table 2: **Scaling Behavior.** SAT's gains amplify with model capacity across diverse architectures and sizes.

Backbone	Method	C10-C	C100-C	TIN-C	IN-C
ViT-B/16	WATT	76.2	49.0	31.7	24.4
	SAT	**80.1**	**51.2**	**37.7**	**28.4**
ViT-L/14	WATT	80.1	54.3	43.3	36.3
	SAT	**86.4**	**62.2**	**50.4**	**38.8**
SigLIP	Baseline	59.0	34.8	22.0	26.5
	SAT	**76.2**	**49.7**	**26.8**	**31.9**

Computational Efficiency

Fig. 3 depicts the running time required for relevant baselines and the proposed SAT across several datasets. These results expose that different TTA methods, particularly SoTA, substantially differ in total runtimes. In particular, recent CLIPArTT and WATT constantly increase the required runtime with the number of classes, driven by their iterative nature. BATCLIP is also highly efficient, yet its runtime still shows a slight increase as the number of classes grows. In contrast, SAT avoids this overhead by distilling this information during the adaptation stage and computing multiple text embeddings *off-line* only once. It is worth noting that the Sinkhorn algorithm used for generating pseudo-codes is highly efficient, accounting for only nearly 1% of the total runtime.

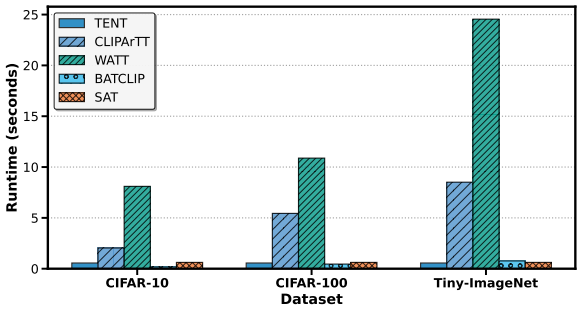

Figure 3: **Inference Runtime**. Seconds per batch ($N = 128$) on NVIDIA A6000.

Batch Size Robustness

In Table 3, we evaluate the sensitivity of SAT to batch size, a common bottleneck for methods utilizing global constraints. While competitive baselines like WATT suffer a significant drop of 7.3% when transitioning from a batch size of 128 to 16, SAT remains remarkably stable, degrading by only 2.9%. Performance for all methods generally improves with a larger batch, confirming that a sufficiently large batch is important for the stability of TTA.

Table 3: **Batch Size Robustness.** Accuracy drop comparison as batch size decreases.

Method	128	32	16
WATT	73.6	69.3	66.3
BATCLIP	68.6	64.2	62.1
SAT	**77.1**	**75.3**	**74.2**

4 Conclusion

We have presented Semantic Anchor Transport (SAT), a novel approach for the test-time adaptation of vision-language models. SAT reformulates TTA as a principled cross-modal alignment problem. It generates robust, batch-aware pseudo-labels by aligning visual embeddings to fixed text-based semantic anchors using Optimal Transport. This global assignment strategy fundamentally mitigates the error accumulation demonstrated by other TTA methods. Furthermore, SAT employs a sophisticated multi-template distillation strategy to harness diverse textual clues, enhancing robustness without incurring significant computational overhead. Extensive experiments demonstrate that SAT achieves state-of-the-art performance on TTA across multiple visual domain shift benchmarks and multi-modal backbones.

References

Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. PLOT: Prompt learning with optimal transport for vision-language models. In *The Eleventh International*

Conference on Learning Representations, 2023.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems (NeurIPS)*, 26, 2013.

Gustavo Adolfo Vargas Hakim, David Osowiechi, Mehrdad Noori, Milad Cheraghalikhani, Ali Bahri, Moslem Yazdanpanah, Ismail Ben Ayed, and Christian Desrosiers. CLIPArTT: Lightweight adaptation of CLIP to new domains at test time. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and surface noise. In *International Conference on Learning Representations (ICLR)*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. In *Preprint*, 2012.

John Lee, Max Dabagia, Eva Dyer, and Christopher Rozell. Hierarchical optimal transport for multimodal distribution alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. doi: 10.48550/arXiv.1906.11768.

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 (ICCV)*, pp. 5542–5550, 2017.

Sarthak Kumar Maharana, Baoming Zhang, Leonid Karlinsky, Rogerio Feris, and Yunhui Guo. BATCLIP: Bimodal online test-time adaptation for clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Xingchao Peng, Ben Usman, Neela Kaushik, Dequan Wang, Judy Hoffman, and Kate Saenko. Visda: A synthetic-to-real benchmark for visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

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 (ICML)*, pp. 8748–8763, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do CIFAR-10 classifiers generalize to cifar-10? In *Preprint*, volume abs/1806.00451, 2018.

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 (NeurIPS)*, 35:14274–14289, 2022.

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 (CVPR)*, pp. 5018–5027, 2017.

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 (ICLR)*, 2021.

Jiayu Wu, Qixiang Zhang, and Guoxi Xu. Tiny imagenet challenge. In *Preprint*, 2017. URL `https://api.semanticscholar.org/CorpusID:212697711`.

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 (CVPR)*, pp. 15922–15932, 2023.

Yunbei Zhang, Akshay Mehra, and Jihun Hamm. OT-VP: Optimal transport-guided visual prompting for test-time adaptation. *arXiv preprint arXiv:2407.09498*, 2024.

A ADDITIONAL DATASETS DETAILS.

Datasets. CIFAR-10.1 Recht et al. (2018) introduces a natural shift from CIFAR-10 Krizhevsky (2012), whereas CIFAR-10-C and CIFAR-100-C are augmented with 15 different corruptions across 5 severity levels Hendrycks & Dietterich (2019) (each containing 10,000 images), leading to 75 corruption scenarios commonly employed in domain shift problems. These datasets are critical for evaluating how well models can generalize to real-world variations and noise that are not present in the training set. Similarly, ImageNet-C Hendrycks & Dietterich (2019) provides these corruptions for the 1000-class ImageNet validation set. Tiny-ImageNet Wu et al. (2017) is a downsized version of the original ImageNet dataset, providing a more accessible challenge with 200 classes. Tiny-ImageNet-C further extends Tiny-ImageNet by incorporating various common corruptions. To assess the performance on class imbalanced datasets, we employ several datasets from Domainbed (PACS Li et al. (2017), OfficeHome Venkateswara et al. (2017), VisDA-C Peng et al. (2018)) that are often utilized for benchmarking domain adaptation algorithms. Each dataset presents unique domain shift challenges and diverse visual categories, allowing for a comprehensive evaluation of a model's ability to generalize across different environments.

CLIP's text templates. In the experimental setup, several predefined text templates from CLIP were used to evaluate the proposed model's adaptability and performance. These are designed to generalize across different contexts and image types. In Table 4, we provide the templates used in our experiments. Each template includes a placeholder, which is dynamically replaced with the class name during the generation of text prompts. Note that, for the sake of fairness with prior literature, the templates are the same as the ones employed in WATT Osowiechi et al. (2024).

Table 4: The different templates used during the experiments.

#	Template
1	"a photo of a {class k}"
2	"itap of a {class k}"
3	"a bad photo of the {class k}"
4	"a origami {class k}"
5	"a photo of the large {class k}"
6	"a {class k} in a video game"
7	"art of the {class k}"
8	"a photo of the small {class k}"

B ADDITIONAL EXPERIMENTAL DETAILS.

B.1 CONFIGURATION: BASELINES

In this section, we detail the configurations used to assess the impact of each main component of our approach, which we refer to *Training-free* OT and SAT with Average Template. *Training-free* OT motivates the use of the soft assignments Q^* over the CLIP baseline, whereas SAT with Average Template includes these assignments to fine-tune the model. Furthermore, we want to highlight that the latter does not fully leverage multiple individual templates, which is introduced in our proposed method. Given a batch of test images, *Training-free* OT (Algorithm 1) computes soft assignments for each text template, which are later averaged to produce a final prediction. On the other hand, SAT with Average Template (Algorithm 2) utilizes the averaged assignments (obtained by *Training-free* OT) to refine the visual encoder. More concretely, at each batch, the average assignments $\tilde{Q}$ supervise the predictions of the test images produced by the model. These predictions are obtained by resorting to Eq. (1), where the average class embedding T is used, together with the visual embeddings of the test images, Z. Differences between the assignments and the predictions are minimized via a cross-entropy loss, whose gradients are used to update the layer norm parameters of the model.

B.2 EXTENDED NUMERICAL VALUES

We further substantiate the findings presented in Figure 3 by providing detailed results in Tables 5 and 6. The tables showcase a comprehensive performance comparison of the impact of the different

Algorithm 1 *Training-free* OT for one test batch.

Input: Test batch $\{\boldsymbol{x}_i\}_{i=1}^{B_T}$, set of M semantic anchor matrices $\{\mathbf{T}_m\}_{m=1}^{M}$, visual encoder $\boldsymbol{\theta}$.
// Pre-compute text prototypes.
$\mathbf{T} \in \mathbb{R}^{d\times K\times M} = [\mathbf{t}_{km}]$
// — Alignment Phase —
for each template m in a random permutation of $\{1,\dots,M\}$ **do**
    // Step 1: Align - Compute soft assignments $\mathbf{Q}_m^*$
    $\mathbf{T}_m \in \mathbb{R}^{d\times K}$
    $\mathbf{Z} \leftarrow [\boldsymbol{\theta}(\boldsymbol{x}_1),\dots,\boldsymbol{\theta}(\boldsymbol{x}_{B_T})]$
    $\mathbf{Q}_m^* \leftarrow \text{SolveOT}(\mathbf{Z}, \mathbf{T}_m)$ Eq. 9
end for
// — Inference Phase —
// Predict by averaging soft assignments
$\mathbf{P} \leftarrow \frac{1}{M}\sum_{m=1}^{M} \mathbf{Q}_m^*$
Return $\mathbf{P}$

Algorithm 2 SAT with Average Template for one test batch.

Input: Test batch $\{\boldsymbol{x}_i\}_{i=1}^{B_T}$, set of M semantic anchor matrices $\{\mathbf{T}_m\}_{m=1}^{M}$, visual encoder $\boldsymbol{\theta}$.
$\mathbf{T} \leftarrow \frac{1}{M}\sum_{m=1}^{M} \mathbf{T}_m$
// — Adaptation Phase —
for each template m in a random permutation of $\{1,\dots,M\}$ **do**
    // Step 1: Align - Compute soft assignments
    $\mathbf{Q}_m^* \leftarrow \text{SolveOT}(\mathbf{Z}, \mathbf{T}_m)$ Eq. 9
end for
// Step 2: Average assignments & refine encoder
$\tilde{\mathbf{Q}} \leftarrow \frac{1}{M}\sum_{m=1}^{M} \mathbf{Q}_m^*$
$\mathbf{P} \leftarrow \text{Predict}(\mathbf{Z}, \mathbf{T})$ Eq. 1
$\mathcal{L} \leftarrow \text{CrossEntropy}(\mathbf{P}, \tilde{\mathbf{Q}})$
Update LayerNorm parameters of $\boldsymbol{\theta}$ using $\nabla_{\boldsymbol{\theta}}\mathcal{L}$.
// — Inference Phase —
$\mathbf{Z} \leftarrow [\boldsymbol{\theta}(\boldsymbol{x}_1),\dots,\boldsymbol{\theta}(\boldsymbol{x}_{B_T})]$
$\mathbf{P} \leftarrow \text{Predict}(\mathbf{Z}, \mathbf{T})$
Return $\mathbf{P}$

Table 5: Numerical Analysis for CIFAR-10, CIFAR-10.1 and CIFAR-10-C as shown in Figure 3.

Dataset	CLIP	TF-OT	Avg.T	SAT
CIFAR-10	88.74	89.44	91.40	**93.15**
CIFAR-10.1	83.25	84.30	85.50	**88.37**
Gaussian Noise	35.27	46.68	54.68	**64.85**
Shot Noise	39.67	49.84	57.13	**67.34**
Impulse Noise	42.61	47.18	52.72	**62.27**
Defocus Blur	69.76	73.22	77.84	**82.09**
Glass Blur	42.40	50.09	56.88	**68.07**
Motion Blur	63.97	70.15	74.81	**81.30**
Zoom Blur	69.83	74.19	77.99	**83.13**
Snow	71.78	73.93	78.41	**83.71**
Frost	72.86	75.66	79.31	**83.40**
Fog	67.04	69.28	75.54	**82.56**
Brightness	81.87	82.90	86.87	**89.90**
Contrast	64.37	67.30	75.30	**84.86**
Elastic Trans.	60.83	64.06	69.54	**76.08**
Pixelate	50.53	56.65	62.91	**76.25**
JPEG Comp.	55.48	59.75	64.31	**70.03**
Mean	59.22	64.05	69.62	**77.06**

Table 6: Numerical Analysis for CIFAR-100 and CIFAR-100-C datasets as shown in Figure 3.

Corruption	CLIP	TF-OT	Avg.T	SAT
CIFAR-100	61.68	61.55	68.26	**71.68**
Gaussian Noise	14.80	21.26	23.29	**33.43**
Shot Noise	16.03	22.98	25.15	**35.60**
Impulse Noise	13.85	22.27	22.14	**30.94**
Defocus Blur	36.74	41.40	46.84	**53.87**
Glass Blur	14.19	22.47	23.60	**35.26**
Motion Blur	36.14	40.71	44.41	**52.77**
Zoom Blur	40.24	45.14	49.97	**56.71**
Snow	38.95	44.14	46.73	**54.30**
Frost	40.56	45.03	47.85	**54.92**
Fog	38.00	40.00	45.45	**53.57**
Brightness	48.18	51.26	57.22	**64.43**
Contrast	29.53	33.69	41.16	**55.01**
Elastic Trans.	26.33	33.02	34.85	**43.79**
Pixelate	21.98	27.72	30.13	**44.51**
JPEG Comp.	25.91	31.69	33.80	**40.83**
Mean	29.43	34.85	38.17	**47.33**

components of our approach, which empirically motivate our model. These results are reported for different corruption benchmarks on the CIFAR-10, CIFAR-10.1, CIFAR-10C, CIFAR-100, and CIFAR-100C datasets, respectively, using a ViT-B/32 backbone.

Table 7: **Performance in visual corruptions benchmarks.** Results using CLIP ViT-B/32. Best method in **bold**, second best underlined.

Method	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixelate	JPEG	Mean ↑
CIFAR-10C																
CLIP [ICLR'21]	35.27	39.67	42.61	69.76	42.40	63.97	69.83	71.78	72.86	67.04	81.87	64.37	60.83	50.53	55.48	59.22
TENT [ICLR'21]	41.27	47.20	48.58	77.12	52.65	71.25	76.20	78.29	79.84	77.39	87.78	79.47	70.00	63.74	62.64	67.56
SAR [ICLR'22]	47.58	50.39	47.19	71.65	49.34	70.27	72.63	71.66	72.82	69.48	82.34	70.54	60.98	48.07	58.48	62.89
VTE [ECCVw'24]	44.40	47.70	42.90	64.90	45.00	66.70	67.00	67.40	64.50	65.30	74.90	53.60	61.20	42.60	50.80	57.26
TPT [NeurIPS'22]	33.90	38.20	37.66	67.83	38.81	63.39	68.95	70.16	72.39	64.31	81.30	62.26	56.43	42.80	53.67	56.80
WATT [NeurIPS'24]	63.84	65.28	58.64	78.94	65.12	77.81	79.32	79.79	80.54	78.53	87.11	81.20	72.66	71.11	67.36	73.82
CLIPArTT [WACV'25]	59.90	62.77	56.02	76.74	61.77	76.01	77.40	77.29	79.20	75.74	86.59	77.82	70.20	66.52	63.51	71.17
BATCLIP [ICCV'25]	50.89	56.01	54.35	76.17	56.11	74.71	76.01	77.74	79.33	75.87	86.46	78.65	68.76	56.41	61.79	68.62
SAT	**64.85**	**67.34**	**62.27**	**82.09**	**68.07**	**81.30**	**83.13**	**83.71**	**83.40**	**82.56**	**89.90**	**84.86**	**76.08**	**76.25**	**70.03**	**77.06**
CIFAR-100C																
CLIP [ICLR'21]	14.80	16.03	13.85	36.74	14.19	36.14	40.24	38.95	40.56	38.00	48.18	29.53	26.33	21.98	25.91	29.43
TENT [ICLR'21]	14.38	17.34	10.03	49.05	3.71	46.62	51.84	46.71	44.90	47.31	60.58	45.90	33.09	26.47	29.89	35.19
SAR [ICLR'22]	22.82	25.10	18.68	44.51	21.78	43.04	47.04	46.75	47.34	44.62	57.00	42.17	31.51	25.09	30.83	36.55
VTE [ECCVw'24]	10.00	10.30	13.30	36.10	20.40	37.90	39.80	42.20	40.80	36.60	45.50	29.20	30.80	17.00	20.70	28.71
TPT [NeurIPS'22]	14.03	15.25	13.01	37.60	16.41	37.52	42.99	42.35	43.31	38.81	50.23	28.09	28.12	20.43	28.82	30.46
WATT [NeurIPS'24]	32.07	34.36	30.33	52.99	32.15	50.53	55.30	52.77	53.79	51.49	63.57	52.76	40.90	40.97	39.59	45.57
CLIPArTT [WACV'25]	25.32	27.90	25.62	49.88	27.89	47.93	52.70	49.72	49.63	48.77	61.27	48.55	37.45	33.88	36.07	41.51
BATCLIP [ICCV'25]	17.25	19.76	18.98	42.20	19.00	40.81	46.59	41.34	40.14	41.56	53.85	34.07	31.38	25.51	28.96	33.43
SAT	**33.43**	**35.60**	**30.94**	**53.87**	**35.26**	**52.77**	**56.71**	**54.30**	**54.92**	**53.57**	**64.43**	**55.01**	**43.79**	**44.51**	**40.83**	**47.33**
Tiny-ImageNet-C																
CLIP [ICLR'21]	7.08	9.41	3.44	21.71	9.12	34.52	27.44	32.51	36.33	25.94	40.15	1.81	30.40	22.78	29.59	22.15
TENT [ICLR'21]	8.01	10.04	4.18	24.53	10.09	36.94	29.48	32.20	35.72	27.46	39.79	2.24	31.92	24.79	30.93	23.22
SAR [ICLR'22]	9.09	10.94	3.65	5.50	1.68	14.02	12.08	20.72	24.62	8.37	32.35	0.71	15.32	12.39	25.35	13.12
VTE [ECCVw'24]	18.63	20.34	4.71	9.62	2.21	30.37	21.68	38.84	40.27	17.41	41.22	0.63	31.64	25.33	37.79	22.71
TPT [NeurIPS'22]	9.29	11.70	4.85	27.56	11.03	38.97	34.29	34.45	37.13	28.89	43.31	3.15	33.88	27.70	33.60	25.32
WATT [NeurIPS'24]	13.02	15.94	6.90	29.91	14.01	41.26	33.96	37.76	39.65	32.13	46.93	3.53	35.01	31.55	36.46	27.87
CLIPArTT [WACV'25]	14.44	17.44	10.37	31.46	15.84	41.34	35.06	36.86	38.20	33.44	46.43	6.24	33.89	34.85	37.32	28.88
BATCLIP [ICCV'25]	11.96	15.48	10.05	31.89	14.76	43.31	39.07	39.02	39.05	31.91	49.06	5.65	32.79	36.63	39.12	29.32
SAT	**21.40**	**24.90**	**17.34**	**35.39**	**21.16**	**46.26**	**40.93**	**42.32**	**44.97**	**38.60**	**53.10**	**11.88**	**40.73**	**41.84**	**42.86**	**34.91**
ImageNet-C																
CLIP [ICLR'21]	11.30	11.58	12.28	20.88	8.92	19.78	17.62	19.92	23.48	25.90	47.34	15.48	17.02	28.00	27.60	20.47
TENT [ICLR'21]	8.00	7.20	9.20	23.04	10.84	22.86	19.04	21.24	23.86	26.54	48.54	18.32	17.90	30.32	29.66	21.10
SAR [ICLR'22]	13.07	15.69	13.92	22.74	14.53	23.41	19.49	22.65	24.89	29.47	48.39	18.88	19.61	31.68	29.07	23.17
VTE [ECCVw'24]	7.12	10.24	9.18	**27.31**	10.27	**26.42**	**27.36**	24.28	26.15	31.22	49.37	13.09	14.18	32.44	31.33	22.66
TPT [NeurIPS'22]	8.94	7.22	7.55	20.47	9.13	21.78	23.92	24.61	21.54	24.98	40.37	15.22	13.18	30.74	24.63	20.01
WATT [NeurIPS'24]	7.76	7.06	8.94	24.16	12.46	25.00	21.52	21.58	24.16	26.62	**49.74**	**21.14**	19.90	32.70	32.16	22.33
CLIPArTT [WACV'25]	14.74	15.10	15.30	10.82	9.02	13.82	12.30	16.96	22.52	19.90	41.78	0.26	12.84	22.80	31.94	18.15
BATCLIP [ICCV'25]	14.84	15.10	15.52	24.42	17.18	25.64	23.08	25.06	25.58	**31.08**	49.66	18.44	22.20	**33.42**	**33.02**	24.95
SAT	**18.98**	**24.54**	**25.04**	25.46	**21.98**	26.34	24.46	**27.78**	**28.74**	29.32	41.98	17.90	**25.70**	26.32	25.68	**26.01**

B.3 Detailed Results

Performance under common visual corruptions. We present a unified evaluation across four standard benchmarks in Table 7, which shows the superiority of SAT in adapting CLIP in the presence of common corruptions against a comprehensive suite of recent TTA methods. Compared to vanilla CLIP, our model brings performance gains of 17.8% (CIFAR-10C) and 17.9% (CIFAR-100C) without requiring additional supervision. These performance gains are similar when compared to other popular TTA methods, e.g., TENT or TPT, with differences ranging from 10% to 21%. While this gap is reduced compared to recent approaches, such as WATT, CLIPArTT, and BATCLIP, the differences remain significant. SAT outperforms the second-best competitor, i.e., WATT, by up to 3.2% in CIFAR-10C (e.g., 5.1% in *pixelate*) and 1.8% in the more challenging scenario of

CIFAR-100C (which contains $\times 10$ classes). Additionally, it achieves significant gains over recent baselines in specific corruptions, e.g., 19.8% compared to BATCLIP in *pixelate* (CIFAR-10C) or 16.3% in *Glass Blur* (CIFAR-100C). Also, compared to CLIPArTT, SAT shows gains of nearly 10.6% on *pixelate* (CIFAR-100C) and 8.1% on *Gaussian Noise* (CIFAR-100C). The gap is even more pronounced on the more difficult Tiny-ImageNet-C testbed, where our 34.9% mean accuracy is 5.6% higher than that of the strongest baseline, BATCLIP. Finally, on the highly challenging 1000-class ImageNet-C benchmark, where many methods show limited gains, SAT again proves its robustness. Specifically, achieves the highest mean accuracy, 26.0%, a promising $+5.5\%$ gain over the CLIP baseline, outperforming all other TTA baselines.

B.4 Additional Details on Epsilon (ϵ)

In the main paper, we stated that ϵ, the entropic constraint weight in Eq. 7 was set to $\epsilon = 0.7$ based on preliminary experiments. However, here in the appendix, we explore the sensitivity of the model to different values of ϵ to assess its impact on accuracy under various corruptions and datasets. As shown in Table 8, we experimented with ϵ values of 0.3, 0.5, 0.7 and 0.9 across CIFAR-10C and CIFAR-100C datasets. Further, we show the values for CIFAR-10 and CIFAR-100 datasets along with averaged performance over corruptions in Figure 4. We can observe that higher ϵ values (0.7 and 0.9) improved stability and performance due to stronger regularization introduced by higher entropic constraints, which prevents degenerate solutions during Sinkhorn normalization. While smaller ϵ values sometimes performed better under mild corruptions, they exhibited significant instability during Sinkhorn normalization. In several cases, the Q matrix became NaN, especially under high noise corruptions, rendering the adaptation process unusable which can be attributed to insufficient regularization. The chosen value ($\epsilon = 0.7$) strikes a balance between stability and performance, yielding robust results across both clean and corrupted datasets.

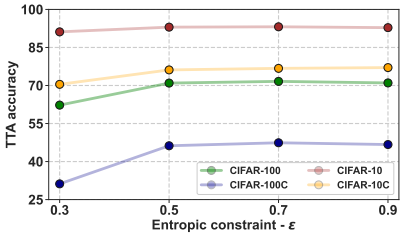

Figure 4: **Ablation on epsilon values** across multiple datasets.

Table 8: Accuracy on CIFAR-10 and CIFAR-100 under different corruptions and epsilon values.

Corruption	**CIFAR-10**				**CIFAR-100**			
	0.3	**0.5**	**0.7**	**0.9**	**0.3**	**0.5**	**0.7**	**0.9**
Original	91.15	93.01	93.15	92.81	62.26	70.95	71.62	71.01
Gaussian Noise	56.16	63.65	64.43	65.26	13.74	32.03	33.2	32.44
Shot Noise	57.53	66.29	66.27	67.4	15.62	34.69	35.33	35.46
Impulse Noise	54.27	61.22	61.84	62.42	19.44	29.94	30.99	30.36
Defocus Blur	76.27	81.33	82.1	82.49	39.16	52.77	54.16	53.32
Glass Blur	57.63	66.75	67.55	67.96	17.06	34.46	35.24	34.89
Motion Blur	71.99	80.11	81.31	80.9	36.59	51.33	52.67	51.62
Zoom Blur	76.24	82.54	83.11	83.46	42.58	55.53	56.93	56.13
Snow	78.59	83.39	82.99	83.56	40.91	53.54	54.06	54.12
Frost	78.92	82.67	82.93	83.41	40.98	54.1	54.93	54.37
Fog	75.75	81.64	81.9	82.47	40.76	52.69	53.6	53.12
Brightness	85.97	89.59	89.72	89.71	48.31	63.27	64.87	63.57
Contrast	70.38	83.55	84.53	84.6	32.09	53.61	55.39	54.41
Elastic Trans.	69.17	75.42	75.67	76.09	27.78	42.69	43.78	43.01
Pixelate	61.31	74.57	76.51	76.08	22.79	42.64	44.96	43.18
JPEG Comp.	66.02	69.42	70.46	70.03	30.16	40.16	41.02	40.67
Mean	70.46	76.14	76.75	76.06	31.20	46.23	47.41	46.71

B.5 Sensitivity to Batch Size

Our method, SAT, is fundamentally a *batch-aware* approach. We evaluate SAT's performance on CIFAR-10C while varying the batch size from 16 to 128, comparing it against other methods. As shown in Table 9, SAT's performance is robust and scales well, achieving strong results even with a

Table 9: Accuracy (%) on **CIFAR-10** and **CIFAR-10C** vs. **Batch Size**. Results are on ViT-B/32.

Dataset	Method	16	32	64	128
CIFAR-10	WATT	89.14	89.51	90.16	91.05
	BATCLIP	84.31	85.22	88.34	90.35
	SAT (Ours)	**87.06**	**92.34**	**93.05**	**93.22**
CIFAR-10C	WATT	65.66	68.34	71.21	73.00
	BATCLIP	63.08	61.89	65.51	69.61
	SAT (Ours)	**74.40**	**77.01**	**77.35**	**77.06**

Table 10: Accuracy (%) of SAT with varying numbers of templates (see Table 4) on natural domain shift benchmarks.

# Templates	1	2	3	4	5	6	7	8
CIFAR-10	91.59	92.16	92.46	92.69	92.88	93.04	93.14	93.14
CIFAR-100	68.36	69.46	70.08	70.56	70.90	71.15	71.38	71.34
Tiny-ImageNet	60.62	61.73	62.32	62.69	62.95	63.16	63.20	63.11

batch size of 32. Notably, under distribution shift on CIFAR-10C, SAT exhibits substantially lower performance degradation when reducing the batch size (only -2.5%, from 77.35 to 74.40) compared to much larger drops observed for WATT (-7.5%) and BATCLIP (-6%), highlighting SAT's superior robustness to batch size constraints in corrupted settings. Performance for all methods generally improves with a larger batch, confirming that a sufficiently large batch is important for the stability of TTA.

B.6 On the impact of multiple templates.

We now assess how the performance evolves as the number of templates increases (Table 10). Concretely, we trained our model multiple times, varying the number of templates $N \in \{1, \ldots, 8\}$. For each N, the average accuracy over all template combinations was computed, leading to 256 experiments per dataset. Results from this experiment show that SAT performance consistently increases with the number of templates. This showcases how leveraging multiple templates individually can extract richer information to adapt CLIP at test time.

C Detailed Experimental Results

To verify that our adaptation method does not simply overfit to corruptions at the cost of performance on the original data distribution, we evaluate SAT on clean, non-corrupted benchmarks. Table 11 presents the performance of SAT and competing baselines on the standard CIFAR-10, CIFAR-10.1, CIFAR-100, and Tiny-ImageNet test sets.

C.1 Performance on Clean Datasets

Table 11 details the performance of all ViT-B/32 methods on the original, uncorrupted test sets. These results correspond to the clean dataset performance reported in the ablations (Figure 3) and form the basis for the corruption-free evaluation. The results show that SAT does not just preserve performance but consistently improves accuracy over the CLIP baseline and other TTA methods, even without a significant domain shift. On CIFAR-10, SAT achieves 93.15% accuracy, surpassing WATT at 91.05%. This trend is even more pronounced in more complex datasets. On CIFAR-100, SAT achieves 71.68%, a full 10-point gain over the 61.68% from CLIP and is notably higher than all other baselines. Similarly, on Tiny-ImageNet, SAT 63.69% provides a clear improvement over both the baseline (58.29%) and WATT (61.35%).

Table 11: Performance on **Clean Datasets** (ViT-B/32). Best method in **bold**.

Dataset	CLIP	TENT	TPT	CLIPArTT	WATT	BATCLIP	SAT
CIFAR-10	88.74	91.69	88.06	90.04	91.05	88.89	**93.15**
CIFAR-10.1	83.25	87.60	81.80	86.35	86.98	83.70	**88.37**
CIFAR-100	61.68	69.74	63.78	69.79	70.74	63.94	**71.68**
Tiny-IN	58.29	57.72	58.90	59.85	61.35	58.93	**63.69**

C.2 Per-Corruption Results

To supplement the mean accuracy results presented in the main paper (Table 1), this section provides a granular, per-corruption performance breakdown. This analysis demonstrates that SAT's strong average performance stems from a consistent robustness across a wide variety of distribution shifts.

We extend this detailed analysis to the larger-scale backbones, ViT-B/16 (Table 12) and ViT-L/14 (Table 13).

On ViT-B/16 (Table 12). SAT achieves the highest accuracy on all 15 corruption types for both CIFAR-10C and CIFAR-100C. For example, on CIFAR-10C *Pixelate* corruption, SAT (78.73%) significantly outperforms the baseline, WATT (75.67%). The gains are similarly strong on ImageNet-C, where SAT (mean 28.35%) consistently outperforms all other methods.

On ViT-L/14 (Table 13). The performance gap widens further, demonstrating SAT's superior scalability. On CIFAR-100C, SAT (mean 69.48%) again dominates every single corruption type. The improvement is validated on challenging corruptions like *Glass Blur*, where SAT (48.19%) more than doubles the performance of the zero-shot baseline (23.46%) and is far ahead of the next-best method, WATT (33.54%). On the difficult Tiny-ImageNet-C, SAT's mean accuracy of 50.36% is a major leap from the 34.98% baseline and all other TTA methods.

This comprehensive, per-corruption breakdown confirms that SAT provides a fundamentally more robust adaptation mechanism that is effective against diverse types of data shifts, and that these benefits are amplified when applied to larger, more capable vision-language models.

D Comparison to OT-VP

The very recent OT-VP Zhang et al. (2024) has presented a solution integrating optimal transport for test-time adaptation. Nevertheless, it presents several fundamental differences with our work. First, OT-VP involves learning a universal visual prompt for the target domain, for which an optimal transport distance is optimized. Thus, optimal transport is used for a different task. Second, OT-VP is tailored to only visual models, not being capable of leveraging the valuable information found on the text modality. And third, the adaptation scenario strongly differs from our setting. In particular, OT-VP first fine-tunes the pre-trained model (pre-trained on ImageNet-1k) to a subdomain (e.g., PACS) with labeled data, to later adapt at test-time to the other subdomains (e.g., OfficeHome). While this strategy can be done in a single-source setting, OT-VP also evaluates the performance when multiple domains are used as the source (e.g., supervised adaptation on PACS), and only one left out for testing, referred to as Multi-Source in Table 14. In contrast, we follow the protocol of recent literature of CLIP test-time adaptation, where CLIP is directly exposed to the unsupervised test data points, without intermediate adaptation steps. To expand the extent of our empirical validation, we compare our approach to OT-VP whose results are reported in Table 14. To ensure a rigorous comparison, we use the same visual encoder as the backbone used in OT-VP and other prior models, i.e., ViT-B/16.

E Discussion on the Two Hyperparameters of WATT and Motivation for SAT

WATT introduces two key hyperparameters: L (number of adaptation iterations for each text embedding) and M (number of repetitions of the weight averaging process). While these hyperparameters play an essential role in improving WATT's adaptation capabilities, they introduce significant computational and scalability challenges, particularly as the number of classes grows. The iterative

Table 12: **Unified corruption robustness results** (ViT-B/16, severity = 5). Top-1 accuracy (%) on CIFAR-10C, CIFAR-100C, Tiny-ImageNet-C, and ImageNet-C. Best method in **bold** (with gray cell), second best underlined.

Method	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixelate	JPEG	Mean ↑
CIFAR-10C																
CLIP	37.75	41.10	51.71	70.07	42.24	65.81	72.50	73.23	76.52	68.35	83.36	61.90	53.16	48.48	56.05	60.15
TENT	31.04	40.54	58.03	77.57	47.16	76.16	79.64	81.68	83.22	80.78	89.85	79.24	62.54	67.08	65.42	68.00
TPT	35.35	41.03	54.86	70.29	37.86	67.43	72.91	72.98	75.87	69.13	83.67	62.16	51.26	44.65	56.73	59.75
WATT	65.57	68.67	70.39	79.90	61.62	79.02	81.10	82.54	83.46	81.88	89.10	83.79	70.93	75.67	69.65	76.22
CLIPArTT	60.89	65.19	67.55	78.92	57.18	76.59	79.62	81.13	81.24	78.47	88.66	75.15	69.49	71.80	66.42	73.22
BATCLIP	61.81	65.49	64.32	80.03	53.82	79.67	81.42	82.63	83.60	80.84	87.20	81.17	69.44	63.17	68.21	73.52
SAT	**68.50**	**71.24**	**76.97**	**83.33**	**68.33**	**83.13**	**85.35**	**86.65**	**86.37**	**85.60**	**92.64**	**86.92**	**75.18**	**78.73**	**72.65**	**80.11**
CIFAR-100C																
CLIP	15.88	17.49	21.43	40.10	13.48	39.82	45.45	42.77	45.39	38.98	52.55	33.32	24.39	21.89	27.21	32.01
TENT	12.28	15.07	13.13	50.35	4.84	49.85	54.76	52.38	51.66	50.74	64.26	48.69	33.56	36.20	30.80	37.90
TPT	15.43	16.88	22.12	41.08	18.43	40.85	46.77	47.24	48.61	39.92	55.83	33.13	27.36	21.26	30.97	33.73
WATT	35.95	37.96	44.62	53.80	33.39	52.72	57.51	56.73	56.48	53.83	66.67	55.06	40.37	47.02	42.13	48.95
CLIPArTT	19.01	20.27	17.66	49.86	18.34	50.00	54.13	52.80	49.56	49.92	63.76	47.86	32.93	39.49	35.56	40.08
BATCLIP	21.79	28.07	30.81	45.46	21.16	45.04	49.82	49.52	47.81	47.32	59.12	43.72	30.89	30.17	32.06	38.85
SAT	**38.80**	**40.89**	**46.39**	**55.50**	**37.89**	**55.10**	**58.77**	**57.84**	**58.09**	**55.68**	**67.76**	**58.29**	**44.32**	**49.46**	**43.87**	**51.24**
Tiny-ImageNet-C																
CLIP	4.80	6.44	4.36	22.23	7.50	33.46	29.86	31.64	33.26	24.26	40.48	1.72	23.85	22.35	27.65	20.92
TENT	13.44	17.46	9.09	32.33	12.47	44.17	39.24	38.91	42.04	31.06	49.26	3.06	34.57	37.73	41.92	29.78
TPT	10.53	12.90	6.57	29.16	10.88	39.91	36.62	38.66	40.95	29.22	47.78	3.58	32.11	29.02	36.51	26.96
WATT	16.78	19.27	10.42	33.33	16.26	44.21	40.26	42.31	44.83	34.08	52.16	5.43	35.11	37.47	42.93	31.66
CLIPArTT	19.25	21.80	13.46	33.49	18.33	45.17	40.53	42.71	44.28	34.78	51.61	7.24	35.16	41.11	44.58	32.90
BATCLIP	11.92	15.36	10.14	31.77	14.80	43.38	39.13	38.92	38.98	31.88	49.11	5.54	32.91	36.72	38.98	29.30
SAT	**24.88**	**27.99**	**20.69**	**37.27**	**22.72**	**48.75**	**45.16**	**45.74**	**48.09**	**41.21**	**55.75**	**13.06**	**41.69**	**45.83**	**46.55**	**37.69**
ImageNet-C																
CLIP	10.70	11.28	10.66	19.60	13.40	18.06	17.54	26.78	27.08	32.86	49.02	14.98	11.94	30.00	29.38	20.89
TENT	7.98	8.68	8.82	21.64	17.08	21.38	19.78	28.26	28.40	33.86	49.54	18.10	13.18	32.56	32.54	22.79
TPT	7.15	6.23	7.03	20.11	11.24	21.36	23.19	26.04	31.17	32.01	50.12	21.05	8.13	39.22	33.15	22.35
WATT	7.68	8.80	9.14	23.70	18.36	24.88	22.54	29.56	29.36	35.52	50.82	21.44	14.60	34.90	34.40	24.38
CLIPArTT	19.10	19.80	19.10	14.10	18.50	19.40	17.00	26.40	29.00	35.20	47.40	2.20	**22.20**	28.10	34.65	23.47
BATCLIP	17.64	19.00	16.50	24.46	19.04	**27.20**	24.20	31.90	29.74	37.46	**52.20**	21.56	18.82	32.12	**35.56**	27.16
SAT	**19.80**	**21.14**	**19.20**	**24.78**	**21.38**	26.82	**24.40**	**32.04**	**32.36**	**38.54**	49.44	**24.42**	20.16	**35.92**	34.84	**28.35**

nature of L requires multiple forward and backward passes for every template, leading to substantial runtime overhead, especially for datasets with many classes like Tiny-ImageNet. Additionally, higher L values, while effective for complex corruptions, risk overfitting to noisy pseudo-labels, while lower L values may lead to under-adaptation. Similarly, M stabilizes adaptation through repeated weight averaging, but its effectiveness diminishes beyond a certain point, with larger values providing negligible performance improvements while exponentially increasing runtime. This dependence on repeated updates for each class and template becomes especially prohibitive for datasets with a large number of classes, as demonstrated in Figure 4, where runtime increases from CIFAR-10 (10 classes) to Tiny-ImageNet (200 classes).

These limitations motivate our proposed approach, SAT, which simplifies and optimizes the adaptation process, achieving robust performance without the drawbacks associated with L and M. SAT eliminates these limitations by leveraging the Sinkhorn algorithm for single-pass optimization and precomputing averaged class embeddings, removing the need for iterative updates and weight averaging. This not only drastically reduces computational overhead but also ensures scalable, robust adaptation across diverse corruption levels and datasets, offering a practical alternative to WATT's hyperparameter-dependent framework.

Table 13: **Unified corruption robustness results** (ViT-L/14, severity = 5). Top-1 accuracy (%) on CIFAR-10C, CIFAR-100C, Tiny-ImageNet-C, and ImageNet-C. Best method in **bold** (with gray cell), second best underlined.

Method	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixelate	JPEG	Mean ↑
CIFAR-10C																
CLIP	64.64	67.82	78.21	80.73	50.29	80.75	82.75	83.01	84.90	78.44	91.67	84.20	65.45	75.10	72.58	76.04
TENT	68.87	71.95	80.22	83.10	57.12	82.69	84.91	85.99	87.15	81.30	93.07	87.93	69.96	78.88	75.49	79.18
TPT	64.44	66.81	76.46	79.01	49.64	78.85	82.32	82.69	84.63	77.56	90.94	82.88	64.81	77.89	71.18	75.01
WATT	72.73	74.60	80.95	83.15	62.35	82.61	85.44	85.61	86.88	81.79	92.59	87.38	71.25	77.67	75.84	80.06
CLIPArTT	70.04	71.44	79.42	81.75	58.13	80.76	83.39	84.48	85.21	79.27	91.87	86.19	67.43	77.88	74.46	78.06
BATCLIP	75.04	79.39	84.19	86.93	71.01	86.80	87.56	89.49	89.44	86.88	94.04	92.00	79.42	84.38	80.32	83.79
SAT	**80.47**	**82.64**	**87.80**	**90.33**	**77.72**	**88.93**	**91.69**	**91.56**	**91.43**	**90.82**	**95.86**	**94.53**	**82.16**	**87.13**	**82.60**	**86.35**
CIFAR-100C																
CLIP	30.55	34.58	44.89	48.88	23.46	50.83	56.02	49.03	53.27	48.51	60.53	50.24	35.07	43.86	39.11	44.59
TENT	36.93	38.23	49.09	55.23	27.02	56.03	61.19	55.60	58.21	49.26	67.34	59.91	38.49	48.37	43.44	50.14
TPT	36.10	40.96	49.69	50.43	24.35	51.94	56.96	54.89	58.15	53.37	66.60	53.64	35.72	44.32	44.42	47.58
WATT	44.13	46.63	56.26	57.66	33.54	57.81	62.74	61.04	62.76	54.70	71.60	63.95	41.27	51.22	49.78	54.34
CLIPArTT	41.46	44.27	51.44	56.55	30.47	56.98	62.56	58.81	60.38	54.38	69.63	63.39	39.57	50.45	47.45	52.52
BATCLIP	39.87	43.50	50.11	51.55	30.95	51.03	55.98	53.77	54.53	49.08	65.52	59.07	34.07	47.96	45.61	48.84
SAT	**52.26**	**54.59**	**63.02**	**65.20**	**48.19**	**65.09**	**68.92**	**66.81**	**67.71**	**65.54**	**76.25**	**72.34**	**52.72**	**59.55**	**54.98**	**62.21**
Tiny-ImageNet-C																
CLIP	17.26	21.57	17.32	35.63	13.62	50.11	43.65	48.04	49.36	37.58	55.85	5.78	34.15	45.90	50.13	34.98
TENT	26.60	28.93	22.22	38.66	10.98	52.53	46.02	49.13	50.46	38.63	58.25	8.66	36.81	50.77	51.88	40.28
TPT	23.22	27.39	19.79	39.44	15.72	53.91	**48.82**	51.50	53.89	40.43	61.07	7.48	40.65	52.28	52.38	41.07
WATT	27.98	31.47	25.38	40.55	17.44	54.46	48.28	52.60	53.77	42.57	62.01	9.21	39.85	51.83	54.19	43.28
CLIPArTT	29.27	33.01	27.93	39.99	17.09	53.63	47.39	51.97	51.49	42.35	60.69	11.25	38.44	52.22	52.46	42.98
BATCLIP	23.98	27.34	23.06	35.40	14.42	48.27	42.72	41.82	46.54	36.29	54.78	6.90	36.84	47.59	49.89	35.72
SAT	**38.14**	**41.02**	**35.48**	**46.03**	**27.93**	**57.78**	**52.67**	**56.85**	**57.56**	**50.49**	**65.28**	**21.57**	**47.86**	**58.23**	**56.82**	**50.36**
ImageNet-C																
CLIP	19.26	21.62	17.88	27.20	20.26	31.78	29.22	43.16	38.08	43.38	61.52	29.50	20.76	44.22	32.94	32.05
TENT	20.48	22.70	15.86	28.00	21.74	33.40	30.44	43.96	38.58	43.92	61.88	32.22	22.22	45.84	35.10	33.09
TPT	19.00	26.00	17.00	28.00	16.00	30.00	30.00	42.00	31.00	41.00	59.00	32.00	15.00	43.00	31.00	30.67
WATT	23.32	26.18	6.06	32.26	27.66	38.54	34.96	**47.14**	41.52	46.46	63.32	38.66	26.92	50.10	41.38	36.30
CLIPArTT	27.10	29.00	**29.20**	21.80	21.30	31.00	28.40	39.60	38.50	41.50	**64.60**	18.40	**30.70**	43.30	**48.00**	34.13
BATCLIP	19.26	22.64	16.50	33.02	29.12	**40.50**	**36.32**	47.08	40.86	48.10	63.28	37.28	27.88	**50.56**	43.64	37.07
SAT	**27.68**	**29.50**	28.74	**34.52**	**30.58**	39.08	35.50	47.08	**42.26**	**48.48**	59.60	**39.84**	30.20	48.42	40.88	**38.82**

Table 14: Performance comparison to OT-VP across PACS and OfficeHome datasets. Best results highlighted in bold, whereas value indicates the difference wrt OT-VP in both single (first value) and multi (second value) source scenarios.

Method	PACS	OfficeHome	Avg.
First Setting: Single Source			
OT-VP-B	69.8	66.9	67.3
OT-VP	73.5	68.1	70.0
Second Setting: Multi Source			
OT-VP-B	87.3	74.3	80.6
OT-VP	87.7	75.1	81.2
Our setting: *No specific source*			
SAT (Ours)	**96.9** (+23.4/+9.2)	**83.4** (+15.3/+8.3)	**86.8** (+16.8/+5.6)

F DIFFERENCES WRT PLOT

Note that despite the apparent similarities, our work presents fundamental differences w.r.t. Prompt Learning with Optimal Transport (PLOT) Chen et al. (2023), which prevents from adding the later in the empirical evaluation. **a) Target task.** PLOT tackles few-shot adaptation, requiring labeled data to adapt no novel tasks, whereas our method studies test-time adaptation, where labels or test samples are not available. Thus PLOT cannot be compared in the TTA setting. **b) Adaptation.** Our approach updates the layer norm parameters of CLIP visual encoder, whereas PLOT optimizes the input text prompts (i.e., it falls into the Prompt Learning category). **c) Use of OT.** PLOT leverages OT between learnable prompts and a set of local visual features, whose transport plan is directly used as predicted probabilities. In contrast, we compute the transportation cost between the whole image features and fixed text embeddings, which is later used as a pseudo-supervision to distill knowledge during test-time adaptation.