MORPHEUS: META TEST-TIME ADAPTATION VIA NEURAL COLLAPSE GEOMETRY

Michal Danilowski[1]* **Alexander Murphy** [1] **Young D. Kwon**[2] **Abhirup Ghosh**[13]
[1]University of Birmingham [2]Samsung AI Center-Cambridge, UK [3]University of Cambridge

ABSTRACT

Test-time adaptation (TTA) algorithms make models more accurate on distribution-shifted unlabeled test data. However, the growing diversity of TTA methods has introduced a new challenge: no single TTA strategy consistently dominates across shifts, and several methods suffer catastrophic failures on certain corruption types. This makes it essential to dynamically choose an appropriate TTA method to adapt a given model on a given test data.

However, this is challenging as the test data is unlabeled and we cannot run all TTA methods to choose the most accurate one due to prohibitive compute wastage. In this paper, we discover that the entropy of the class distributions and geometric characteristics of the embedding space produced by the source model on unlabeled test data can predict the most accurate TTA method and post-adaptation accuracy without adapting the model. We empirically show that such a selection of the most accurate method can prevent catastrophic failures by choosing an appropriate method. Further, our method can predict the post-adaptation accuracy with an average RMSE of 0.054 over TTA methods. We believe this will encourage discussions around efficient use of existing TTA methods.

1 INTRODUCTION

Machine learning models deployed in real-world settings frequently encounter distribution shifts degrading accuracy. Test-time adaptation (TTA) addresses this challenge by adapting the model parameters using unlabeled test data at inference time, and has shown promising improvements across a variety of domains (Murphy et al., 2025; Dong et al., 2025; Dang et al., 2026).

However, as literature expands with more TTA algorithms Liang et al. (2025), there is no universally most accurate TTA strategy across all types of distribution shifts and choices of samples. This is because the quality of an adapted model depends on the choice of samples and the order in which they are processed by the model. This affects $i)$ relative ranking of different TTA methods and $ii)$ make several high performing TTA methods catastrophically fail Zhao et al. (2023). This is shown in Figure 1(c) where the adapted model suffers from catastrophic failure after certain batch depending on the ordering the batch.

Therefore, here we propose a meta TTA method, `MORPHEUS` (Figure 1(a)) to predict and adapt the source model using the most accurate TTA method given the target data at hand. Given oracle selection of the best performing method to adapt, this simple algorithm will always be the most accurate in the target data and is expected to be more accurate across data distributions on average. One critical constraint here is that we cannot run each adaptation method and measure the accuracy; Firstly because it will waste a large amount of compute to adapt using all methods, and secondly, the target data is unlabeled which makes accuracy estimation non-trivial. This is the fundamental technical challenge we solve: how to predict post-adaptation accuracy using different TTA algorithms without performing the adaptation.

Prior works on approximation of label-free accuracy estimation (Platanios et al., 2017; Lee et al., 2024b) do not address the problem of predicting post-adaptation performance before adaptation is applied. To the best of our knowledge, this is the first attempt to fill this gap in the literature. Another strand of literature that is related deals with ranking the models based on their transferability in the

*Correspondence to: `mxd411@student.bham.ac.uk`

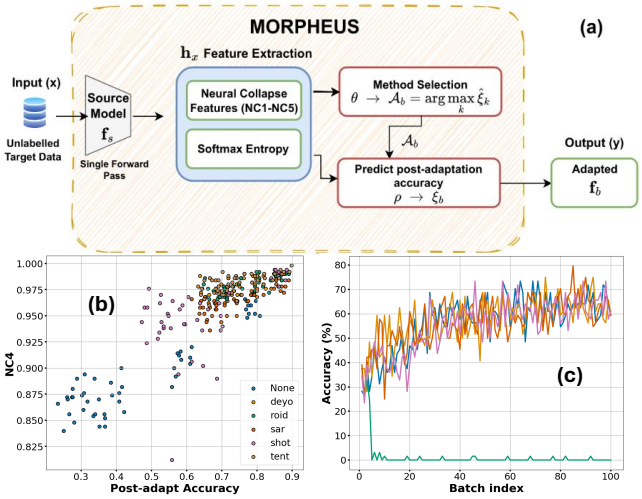


Figure 1: **(a)** High-level overview of MORPHEUS. **(b)** Post-adaptation accuracy across TTA methods correlate with NC4 features extracted from source model embeddings. Shown for Resnet-18 on CIFAR10-C (Gaussian noise; sev 5). **(c)** Different ordering of the batches triggers catastrophic failures in DEYO on ImageNet-C Contrast sev-5. We show five different orderings with the same set of samples.

context of transfer learning Zheng et al. (2025). However, both our goal and the setting differ from that: we aim to find the best adaptation method as opposed to selecting the best model to use in transfer learning, and the unlike our setting, transferability assumes access to labeled target data.

Here we propose a two step method: $i)$ to select the most accurate TTA method to be applied given a target data and source model, and $ii)$ to estimate post-adaptation accuracy for the selected TTA method. We achieve both by using simple regression models. We utilize geometric features (based on Neural Collapse - NC) extracted from embedding vectors when the source model evaluate the target data and entropy of the class probabilities. Figure 1(b) shows that the post adaptation accuracy closely correlate with a feature (details in Appendix) extracted from source embeddings. Such features are efficient to compute as it only needs one forward passes on target data, without performing adaptation itself. Below are our contributions:

1. We introduce MORPHEUS, a single forward-pass-only method for predicting post-adaptation accuracy based on neural-collapse–inspired feature statistics.
2. We show that MORPHEUS can be used as a meta test-time adaptation strategy that prevents catastrophic failures and yields more stable performance across diverse corruptions, achieving accuracy comparable to the benchmark method (65.1% vs. 65.2%) while exhibiting lower variance (0.7 vs. 0.9 std). Furthermore, we find that NC-based post-adaptation prediction is more accurate and stable than entropy-based prediction, particularly in low-accuracy regimes.

2 Problem and Methodology

Consider that a classification model f_s, trained on a labeled source dataset, $D_s = \{X, y\}$ is being tested on unlabeled target data, $\tilde{D} = \{\tilde{X}\}$. Accuracy of f_s drops on $\tilde{D}$ when the distribution shifts from source to target. A test time adaptation method, $\mathcal{A}_m$ uses the target data $\tilde{X}$ to produce the adapted model f_m which is expected to be more accurate on $\tilde{X}$ compared to f_s.

Given f_s and $\tilde{X}$, multiple TTA methods could potentially be applied to adapt. The critical question is how to choose the method that'll produce the highest accuracy. The following are the questions we answer in this paper:

RQ1. Given f_s and $\tilde{X}$ and a set of k TTA methods, select the method $\mathcal{A}_b$ that will produce the highest accuracy on $\tilde{X}$.

RQ2. Considering we decided to use a TTA method, $\mathcal{A}_i$ to adapt a model f_s on dataset $\tilde{X}$, predict the post-adaptation accuracy, $\hat{\xi}$ of f_i on $\tilde{X}$.

The goal is to answer both the questions prior to adapting the model enabling efficiency.

Feature extraction. We first extract the average entropy of the class probabilities while $\tilde{X}$ is evaluated in f_s. Simultaneously, we extract the embedding vectors, $h_{\tilde{X}}$ from the penultimate layer in f_s, i.e., activations from the layer just before the classifier.

Next we compute geometric features from $h_{\tilde{X}}$, following the Neural Collapse (NC) phenomena Papyan et al. (2020). While Neural Collapse is popularly studied to formalize the convergence properties of the embedding space, here we utilize them to characterize the input distribution shift. We utilize 5 NC features including the original 4 measures proposed by Papyan et al. (2020) and the fifth proposed in Ammar et al. (2023). For example, NC1 measures the L_1 distance of the embedding vectors from the origin, and NC3 measures the distance between the class prototypes and the parameters of the classifier layer. As the NC features require labeled data, we approximate this in $\tilde{D}$ using the pseudo labels produced by f_s.

Algorithms. We use two regression models, θ and ρ, to solve the RQ1 and RQ2, respectively. We select the TTA method with the highest predicted accuracy as the method to be used to adapt. In this paper, all experiments use simple random forest regressors.

We assume that data from a subset ($\mathcal{D}_{tr}$) of all possible domains is available prior to deployment. We use the source model (f_s), to extract softmax entropy and NC-based features using the data in $\mathcal{D}_{tr}$. Next, learn a regression model to predict the post-adaptation accuracy given a TTA method. In the training, we adapt f_s using all available TTA methods. Once a regressor is learned, we test in the rest of the $\mathcal{D}_{te}$ domains in a dataset. Below, we show results for using softmax entropy (Our (Ent.)) and NC-based (Our (NC)) features separately.

3 EXPERIMENTS

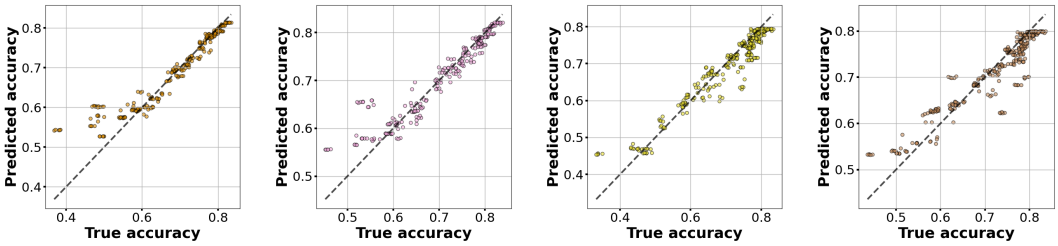
We evaluate MORPHEUS using popular TTA methods (details in Appendix) with hyperparameters as in original papers. We evaluate on ImageNet-C and CIFAR-10-C (appendix) using base version of Vision Transformer (ViT) (Dosovitskiy et al., 2021) and ResNet-18 respectively.

Corruption	No Adapt	SAR	TENT	DEYO	ROID	CoTTA	EATA	FOA	Surgeon	SPA	Our (NC)	Our (Ent.)
Noise												
Gaussian	55.3 (0.4)	58.0 (0.2)	57.5 (0.2)	58.2 (0.3)	59.9 (0.4)	54.3 (7.7)	58.0 (0.3)	57.3 (0.5)	59.1 (0.3)	**61.0 (0.5)**	59.5 (0.3)	**61.0 (0.5)**
Shot	56.0 (0.3)	59.2 (0.4)	58.3 (0.3)	59.4 (0.4)	61.4 (0.4)	58.8 (2.1)	59.2 (0.3)	59.0 (0.3)	60.3 (0.3)	**62.4 (0.4)**	**62.4 (0.4)**	**62.4 (0.4)**
Impulse	56.0 (0.3)	59.4 (0.2)	58.6 (0.2)	59.5 (0.1)	61.2 (0.4)	53.4 (8.9)	59.3 (0.3)	58.0 (0.3)	60.6 (0.2)	**62.5 (0.5)**	61.3 (1.2)	**62.5 (0.5)**
Blur												
Defocus	46.3 (0.4)	51.9 (0.6)	49.0 (0.7)	52.7 (0.4)	**56.8 (0.4)**	48.9 (2.6)	51.8 (0.6)	45.2 (0.5)	55.1 (0.5)	54.6 (1.0)	52.7 (3.2)	54.6 (1.0)
Glass	34.4 (0.4)	45.1 (0.8)	37.7 (0.7)	45.8 (0.6)	52.3 (0.3)	38.5 (0.5)	44.1 (0.4)	33.8 (0.6)	45.7 (0.5)	**55.0 (0.2)**	**55.0 (0.2)**	**55.0 (0.2)**
Motion	52.5 (0.6)	57.1 (0.7)	54.6 (0.7)	57.5 (0.7)	60.8 (0.6)	56.4 (0.7)	56.8 (0.8)	51.5 (0.5)	58.8 (0.7)	**63.8 (0.3)**	60.3 (0.6)	**63.8 (0.3)**
Zoom	43.8 (0.3)	50.0 (0.3)	46.7 (0.5)	50.3 (0.2)	55.4 (0.4)	47.9 (1.3)	49.4 (0.4)	43.4 (0.3)	51.8 (0.2)	**60.1 (0.3)**	54.8 (0.4)	**60.1 (0.3)**
Weather												
Snow	61.8 (0.6)	62.7 (0.6)	62.2 (0.6)	63.2 (0.7)	66.1 (0.9)	65.6 (0.5)	62.9 (0.6)	64.4 (1.0)	64.2 (0.6)	**69.9 (0.5)**	**69.9 (0.5)**	**69.9 (0.5)**
Frost	62.3 (0.8)	58.2 (1.0)	58.5 (0.8)	58.5 (0.9)	62.8 (0.5)	64.8 (0.7)	58.5 (0.7)	61.3 (2.4)	60.9 (0.8)	**69.0 (0.6)**	66.9 (3.0)	**69.0 (0.6)**
Fog	65.0 (0.6)	56.0 (15.0)	47.3 (1.4)	62.5 (1.1)	67.8 (0.6)	65.2 (2.5)	62.6 (0.7)	68.5 (0.5)	37.5 (15.6)	**70.4 (0.6)**	67.8 (0.6)	67.8 (0.6)
Brightness	77.2 (0.4)	77.5 (0.5)	77.2 (0.4)	77.7 (0.4)	78.3 (0.4)	77.7 (0.4)	77.7 (0.4)	76.9 (0.4)	78.4 (0.4)	**79.8 (0.4)**	**79.8 (0.4)**	**79.8 (0.4)**
Digital												
Contrast	31.9 (0.4)	18.1 (18.6)	49.4 (0.5)	45.5 (24.7)	**62.9 (0.6)**	35.0 (10.7)	54.4 (0.4)	47.4 (0.9)	53.5 (0.5)	57.6 (6.9)	60.3 (3.8)	60.3 (3.8)
Elastic	45.3 (0.2)	50.4 (0.5)	47.4 (0.4)	51.4 (0.3)	58.3 (0.5)	50.2 (0.7)	50.3 (0.3)	47.6 (0.6)	52.8 (0.6)	**66.3 (0.8)**	**66.3 (0.8)**	**66.3 (0.8)**
Pixelate	66.4 (0.3)	67.8 (0.3)	67.1 (0.4)	68.4 (0.4)	70.1 (0.3)	69.0 (1.1)	67.9 (0.4)	65.8 (0.2)	69.8 (0.3)	**74.7 (0.2)**	73.4 (1.3)	72.4 (0.3)
JPEG	66.3 (0.3)	67.9 (0.4)	67.1 (0.4)	68.1 (0.4)	68.8 (0.3)	68.3 (0.4)	67.8 (0.4)	66.5 (0.7)	68.6 (0.3)	**71.6 (0.3)**	70.5 (0.8)	**71.6 (0.3)**
Avg.	54.7 (0.4)	56.0 (2.7)	55.9 (0.5)	58.6 (2.1)	62.9 (0.5)	56.9 (2.7)	58.7 (0.5)	56.4 (0.6)	58.5 (1.5)	**65.2 (0.9)**	64.1 (1.2)	65.1 (0.7)

Table 1: Accuracy(%) with std-dev across corruption types and TTA methods with ViT-Base on ImageNet-C (sev-5). Each domain use 6400 samples to adapt. Highest accuracy per corruption type is bold, and the second-highest is underlined. We report average for a $3-$fold cross validation (individual fold are available in the Appendix. MORPHEUS on average is the most accurate.

Table 1 shows that across methods, adaptation consistently improves performance over the no-adaptation baseline, yielding an average gain of $+8.5$ points (from 54.7% to 63.2%). ROID achieves strong performance on most blur and digital corruptions, while SPA performs well under noise and weather corruptions. Our (Ent.) method achieves the highest overall average accuracy (63.2%), outperforming the second-best prior method ROID (62.9%). In particular, it attains either the best or second-best results on the majority of corruption types. Compared to existing methods, our approach demonstrates improved robustness and more stable performance, reflected by consistently high accuracy and relatively low standard deviation.

Figure 2 illustrates the relationship between true and predicted post-adaptation accuracy across domains or severity levels for representative test-time adaptation methods. Across all methods, predicted accuracy closely follows true accuracy, with most points concentrated near the identity line, indicating a strong linear relationship. However, the degree of alignment varies across methods, with noticeable dispersion for lower-performing regimes, particularly at lower true accuracies. This observation is quantitatively supported by the results in Table 2, which report prediction quality using R^2, MAE, and RMSE under both NC and entropy-based settings. Overall, NC-based prediction yields consistently higher R^2 values and lower errors across all methods, indicating more reliable accuracy estimation. Among the evaluated approaches, TENT, Surgeon, and EATA achieve the strongest predictive performance under the NC setting, while entropy-based prediction remains substantially more challenging, leading to reduced correlation and increased error.



(a) Tent (b) Surgeon (c) FOA (d) EATA

Figure 2: True vs predicted post-adaptation accuracy (using NC stats). They correlate in both random partitioning (a, b), and severity level-wise partitioning (c, d).

Method	R^2 (NC)	R^2 (Ent.)	MAE (NC)	MAE (Ent.)	RMSE (NC)	RMSE (Ent.)
SAR	0.697	0.227	0.027	0.057	0.065	0.100
TENT	0.844	0.331	0.024	0.053	0.041	0.084
DEYO	0.625	0.132	0.029	0.054	0.062	0.092
ROID	0.689	0.262	0.027	0.042	0.041	0.062
CoTTA	0.646	0.047	0.035	0.061	0.062	0.099
EATA	0.757	0.310	0.027	0.050	0.045	0.075
FOA	0.662	0.033	0.035	0.065	0.063	0.103
Surgeon	0.749	0.390	0.025	0.049	0.047	0.076
SPA	0.549	0.063	0.029	0.041	0.046	0.066
Avg.	0.679	0.189	0.029	0.053	0.054	0.086

Table 2: Average performance of different methods on NC and Entropy MORPHEUS settings. Across methods, entropy-based predictions have higher errors than NC based prediction

4 Conclusions

We framed test-time adaptation as a decision problem: whether to adapt at all and, if so, which adaptation strategy to apply. By addressing method selection and post-adaptation accuracy prediction, our work enables anticipating adaptation outcomes before incurring computational cost or risking performance degradation. This perspective highlights that inappropriate adaptation can be ineffective or harmful, and that test-time adaptation should not be applied blindly. Our results demonstrate the importance of decision-aware test-time adaptation and point toward more reliable and stable deployment of adaptive models under distribution shift.

REFERENCES

Mouïn Ben Ammar, Nacim Belkhir, Sebastian Popescu, Antoine Manzanera, and Gianni Franchi. Neco: Neural collapse based out-of-distribution detection. *arXiv preprint arXiv:2310.06823*, 2023.

Christina Baek, Yiding Jiang, Aditi Raghunathan, and J Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. *Advances in Neural Information Processing Systems*, 35:19274–19289, 2022.

Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. *Advances in Neural Information Processing Systems*, 34:14980–14992, 2021.

Ting Dang, Soumyajit Chatterjee, Hong Jia, Yu Wu, Flora Salim, and Fahim Kawsar. Adanodes: Test time adaptation for time series forecasting using neural odes. *arXiv preprint arXiv:2601.12893*, 2026.

Jiaheng Dong, Hong Jia, Soumyajit Chatterjee, Abhirup Ghosh, James Bailey, and Ting Dang. Ebats: Efficient backpropagation-free test-time adaptation for speech foundation models. *arXiv preprint arXiv:2506.07078*, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1134–1144, 2021.

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*, pp. 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Wanli Hong and Shuyang Ling. Neural collapse for unconstrained feature model under cross-entropy loss with imbalanced data. *Journal of Machine Learning Research*, 25(192):1–48, 2024. URL http://jmlr.org/papers/v25/23-1215.html.

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of sgd via disagreement. In *International Conference on Learning Representations*.

Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=9w3iw8wDuE.

Taeckyung Lee, Sorn Chottananurak, Taesik Gong, and Sung-Ju Lee. Aetta: Label-free accuracy estimation for test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28643–28652, 2024b.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

Ke Ma, Jiaqi Tang, Bin Guo, Fan Dang, Sicong Liu, Zhui Zhu, Lei Wu, Cheng Fang, Ying-Cong Chen, Zhiwen Yu, and Yunhao Liu. Surgeon: Memory-adaptive fully test-time adaptation via dynamic activation sparsity. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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

John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International conference on machine learning*, pp. 7721–7735. PMLR, 2021.

Alexander Murphy, Michal Danilowski, Soumyajit Chatterjee, and Abhirup Ghosh. Neo: No-optimization test-time adaptation through latent re-centering. *arXiv preprint arXiv:2510.05635*, 2025.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

Shuaicheng Niu, Chunyan Miao, Guohao Chen, Pengcheng Wu, and Peilin Zhao. Test-time model adaptation with only forward passes. *arXiv preprint arXiv:2404.01650*, 2024.

Shuaicheng Niu, Guohao Chen, Peilin Zhao, Tianyi Wang, Pengcheng Wu, and Zhiqi Shen. Self-bootstrapping for versatile test-time adaptation. In *The International Conference on Machine Learning*, 2025.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020. doi: 10.1073/pnas.2015509117. URL https://www.pnas.org/doi/abs/10.1073/pnas.2015509117.

Emmanouil Platanios, Hoifung Poon, Tom M Mitchell, and Eric J Horvitz. Estimating accuracy from unlabeled data: A probabilistic logic approach. *Advances in neural information processing systems*, 30, 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*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2022.

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. *arXiv preprint arXiv:2306.03536*, 2023.

Yaoyan Zheng, Huiqun Wang, Nan Zhou, and Di Huang. Implicit modeling for transferability estimation of vision foundation models. *arXiv preprint arXiv:2510.23145*, 2025.

A Implementation Details

A.1 Morpheus

Algorithm 1 Morpheus pseudo-code

Require: Pretrained classifier $f : \mathbb{R}^m \to \mathbb{R}^C$, unlabeled target data $\tilde{X}$
Require: Candidate TTA methods $\{\mathcal{A}_k\}_{k=1}^K$
Require: NC feature extractor $\Phi_{\text{NC}}(\cdot)$, predictor $g(\cdot)$
Ensure: Selected adaptation method $\mathcal{A}^\star$

```
_NC ← Φ_NC(f, X̃)                          ▷ NC features from forward passes of f
for each method A_k do
    ξ̂_k ← g(_NC, A_k)                     ▷ ξ̂_k ≈ ξ(A_k(f, X̃), X̃, Ỹ)
end for
A* ← arg max_{A_k} ξ̂_k
return A*
```

B Related Work and Background

B.1 Accuracy estimation

Prior work on label-free accuracy estimation aims to infer test accuracy from unlabeled data using confidence or agreement statistics (e.g., Platanios et al. (2017)). Most existing accuracy estimation methods concentrate on ensembles of pretrained models ((Chen et al., 2021; Guillory et al., 2021; Jiang et al.; Miller et al., 2021; Baek et al., 2022). Among them, Agreement-on-the-Line (Baek et al., 2022) and Accuracy-on-the-line (Miller et al., 2021) show a clear linear correlation between model performance across various architectures and distribution shifts , based on the consistency of predictions between in-distribution (ID) and out-of-distribution (OOD) data. The self-training ensemble method (Chen et al., 2021) evaluates the accuracy of a pretrained classifier through an iterative process that trains an ensemble on labeled training data, unlabeled test data, and incorrectly predicted samples. Difference of Confidence (DoC) (Guillory et al., 2021) uses the disparity in model confidence between ID and OOD samples to infer the accuracy gap caused by distribution shifts and derive the final OOD accuracy. Recently, AETTA (Lee et al., 2024b) proposed entropy and consistency-based estimators to monitor accuracy during test-time adaptation. However, these approaches estimate the performance of a fixed or current model state and are not designed to predict or compare post-adaptation accuracy across different TTA strategies, which is the focus of our work.

B.2 Neural Collapse

Neural collapse is a phenomenon observed when training a model after having reached near-zero training loss (Papyan et al., 2020). It describes the geometric properties of the last fully connected layer weights and embeddings fed into the last layer. The behavior occurs, because large neural networks can be interpreted as an unconstrained feature model (Hong & Ling, 2024), where any output can be approximated by the feature extractor, and thus outputs that minimize the norm of the weights are chosen, resulting in neural collapse. More specifically, the four properties are observed. In below definitions, $\boldsymbol{\mu}_G$ is the global mean of the embeddings of the training data, and $\boldsymbol{\mu}_c$ is the class-specific mean embedding for class c in the training set. We also define the within-class covariance to be $\boldsymbol{\Sigma}_W = \text{Avg}_{i,c}\{(\boldsymbol{h}_{i,c} - \boldsymbol{\mu}_c)(\boldsymbol{h}_{i,c} - \boldsymbol{\mu}_c)^T\}$, where $\boldsymbol{h}_{i,c}$ is the embedding of sample $\boldsymbol{x}^{(i)}$ from class c. We also define $\tilde{\boldsymbol{\mu}}_c = (\boldsymbol{\mu}_c - \boldsymbol{\mu}_G) / \|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|_2$, $\boldsymbol{M} = [\boldsymbol{\mu}_c - \boldsymbol{\mu}_G, c = 1, ..., C] \in \mathbb{R}^{p \times C}$, $\boldsymbol{W}$ contains the last-layer weights and $\delta_{c,c'}$ is the Kronecker delta symbol (Papyan et al., 2020).

- **(NC1) Variability collapse:** Near-zero variation in features within same class.

$$\boldsymbol{\Sigma}_W \to 0$$

- **(NC2) Convergence to simplex equiangular tight frame (ETF):** The class means of the features form the vertices of an ETF simplex.

$$\left| \|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|_2 - \|\boldsymbol{\mu}_{c'} - \boldsymbol{\mu}_G\|_2 \right| \to 0 \quad \forall c, c'$$

Method	Noise			Blur				Weather				Digital				Average
	Gauss.	Shot	Impul	Defoc	Glass	Motion	Zoom	Snow	Frost	Fog	Brit	Contr	Elastic	Pixel	JPEG	Acc
Source	55.3	56.2	56.0	46.5	34.8	52.9	44.2	62.4	62.7	65.6	77.7	32.0	45.7	66.7	66.7	55.0
TENT	59.7	61.2	61.2	55.8	50.0	60.1	54.1	64.0	60.6	16.0	78.5	63.3	53.7	69.9	68.7	58.4
FOA	58.4	60.4	58.7	48.7	39.2	54.6	47.5	65.2	60.2	67.2	77.3	51.2	50.3	68.9	65.6	58.2
Surgeon	**63.4**	**65.3**	64.4	**61.7**	63.3	68.4	15.9	8.4	70.4	**75.9**	**81.0**	**70.8**	73.7	76.9	74.5	62.3
EATA	60.8	62.0	61.8	58.4	56.6	63.1	58.9	67.6	64.8	69.9	79.3	65.9	63.1	72.8	70.5	65.0
SAR	59.9	61.3	61.2	57.4	56.1	61.7	57.8	66.0	63.4	65.4	78.8	57.5	61.7	72.1	69.9	63.3
DeYO	59.7	61.1	60.8	57.7	56.8	62.5	58.3	67.4	64.8	69.3	79.3	65.5	64.7	73.3	70.7	64.8
ROID	62.0	63.3	63.0	60.6	59.0	64.9	61.1	69.8	67.8	73.2	79.7	67.8	67.0	74.3	71.8	67.0
SPA	63.3	**65.3**	**64.5**	61.2	**63.5**	**69.4**	**67.7**	**74.2**	**72.7**	75.6	80.8	65.5	**74.0**	**77.7**	**74.8**	**70.0**

Table 3: Comparisons of state-of-the-art methods on ImageNet-C (severity level 5) with ViT-Base regarding Accuracy (%)

$$\langle \tilde{\mu}_c, \tilde{\mu}_{c'} \rangle \to \frac{C}{C-1}\delta_{c,c'} - \frac{1}{C-1} \quad \forall c, c'$$

- **(NC3) Convergence to self-duality:** The class means and last-layer weights converge to each other, upon rescaling.

$$\left\| \frac{\boldsymbol{W}^T}{\|\boldsymbol{W}\|_F} - \frac{\boldsymbol{M}^T}{\|\boldsymbol{M}\|_F} \right\|_F \to 0$$

- **(NC4) Simplification to nearest class-center:** During inference, the last layer assigns the class, by selecting the class mean with the lowest euclidean distance from the sample embedding.

$$\arg\max_{c'} \langle \boldsymbol{m}_c, h(\boldsymbol{x}) \rangle + b_{c'} \to \arg\min_{c'} \|h(\boldsymbol{x}) - \boldsymbol{\mu}_{c'}\|_2$$

C EXPERIMENTAL SETTINGS

We evaluate our method with popular adaptation methods: SAR (Niu et al., 2023), TENT (Wang et al., 2021), CoTTA (Wang et al., 2022), FOA (Niu et al., 2024), Surgeon (Ma et al., 2025), DEYO (Lee et al., 2024a), ROID (Marsden et al., 2024), EATA (Niu et al., 2022) and SPA (Niu et al., 2025). The hyperparameters used are taken directly from their original paper. We use a batch size of 64 for all experiments. We evaluate on ImageNet-C (50 samples $\times$ 1000 classes $\times$ 15 corruption types) (Hendrycks & Dietterich, 2019). We additionally evaluate on CIFAR-10C, which contains 10 classes and 19 corruption types with 5 severity levels. We also consider case where not all samples are used for adaptation. In our experiments we use base version of Vision Transformer (ViT) (Dosovitskiy et al., 2021) and ResNet-18 (He et al., 2016).

D MORE RESULTS

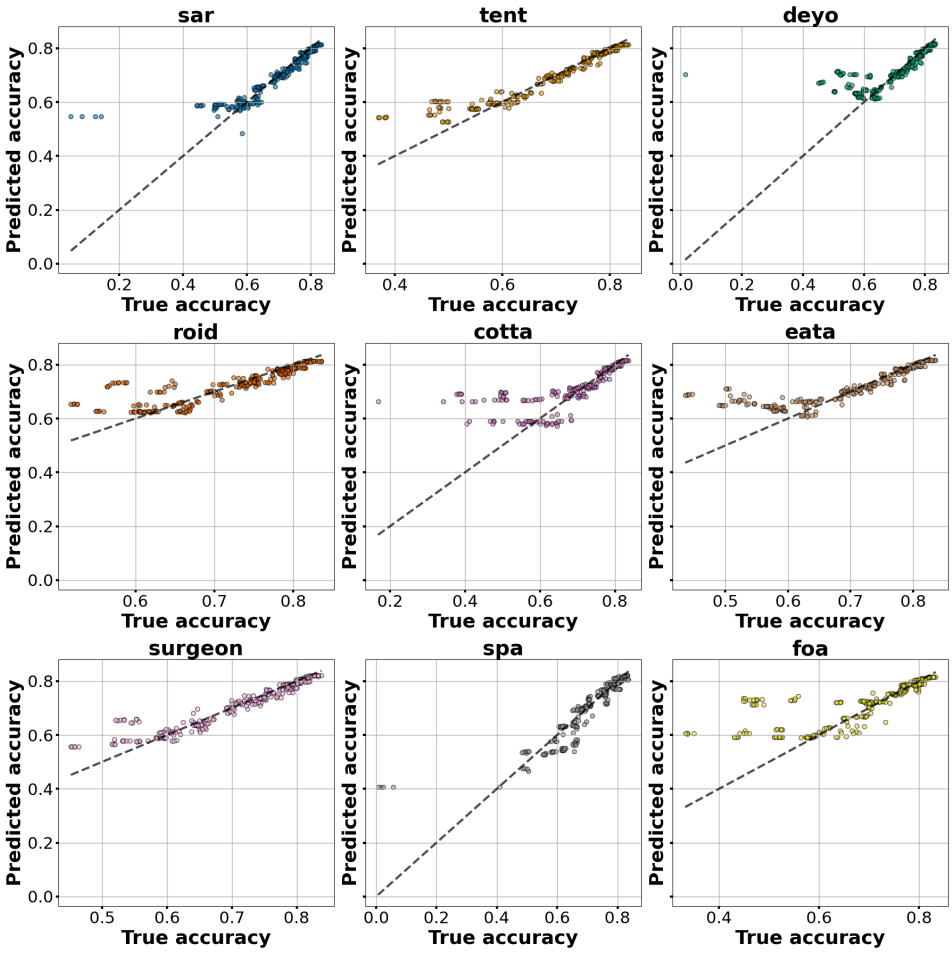


Figure 3: Predicting accuracy of ViT-Base based on source model NC stats (5 domains as training data and 10 as test)

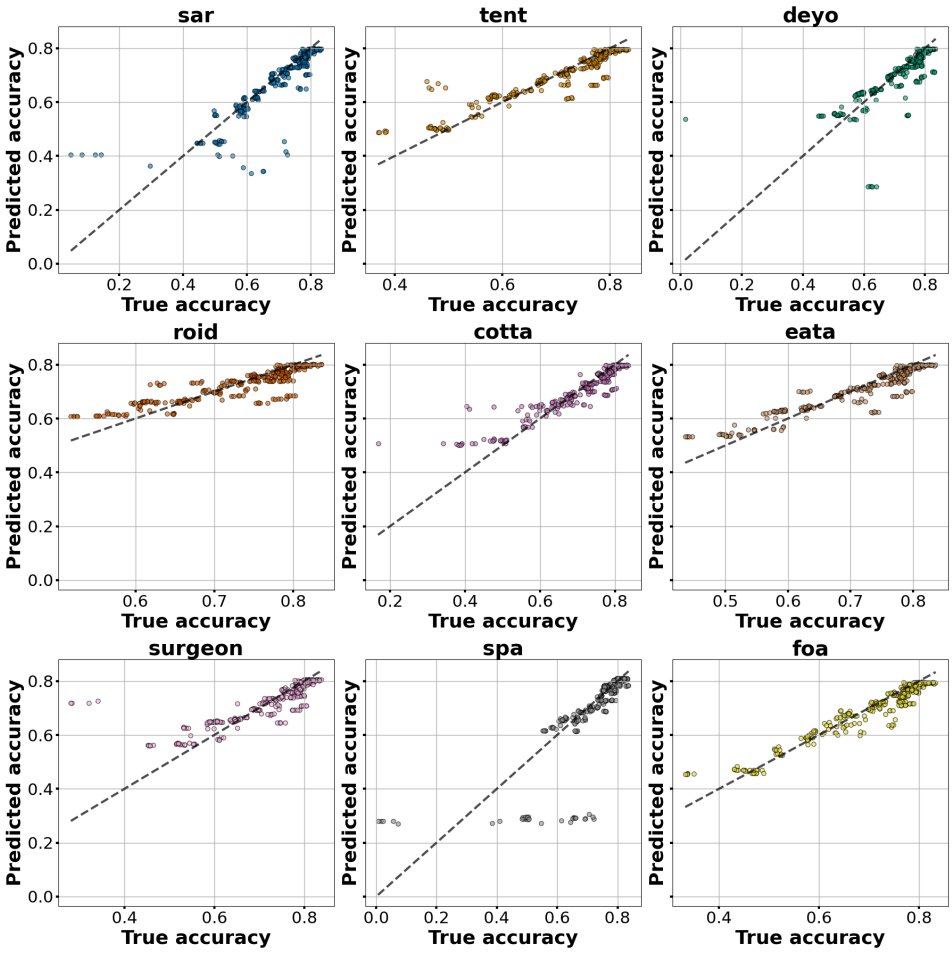


Figure 4: Predicting accuracy of ViT-Base based on source model NC stats (severity 4 samples as training, all remaining as test)

Corruption	No Adapt	SAR	TENT	DEYO	ROID	Our (NC)
Noise						
Gauss.	26.9	65.5 (0.0)	67.4 (0.0)	68.4 (0.3)	**69.9 (0.3)**	68.4 (0.3)
Impulse	19.2	60.0 (0.0)	62.9 (0.0)	**63.6 (0.2)**	63.5 (0.2)	**63.6 (0.2)**
Shot	34.8	68.0 (0.0)	70.8 (0.0)	72.0 (0.3)	**72.1 (0.5)**	72.0 (0.3)
Spatter	75.1	81.9 (0.0)	83.2 (0.0)	**84.2 (0.1)**	83.4 (0.2)	**84.2 (0.1)**
Speckle	40.2	67.8 (0.0)	**72.5 (0.0)**	72.5 (0.4)	72.0 (0.1)	**72.5 (0.4)**
Blur						
Gauss.	37.1	85.6 (0.0)	86.7 (0.0)	**87.1 (0.2)**	85.7 (0.2)	85.6 (0.0)
Glass	49.9	63.3 (0.0)	65.2 (0.0)	**65.9 (0.2)**	65.8 (0.2)	**65.9 (0.2)**
Zoom	64.0	86.1 (0.0)	87.2 (0.0)	**87.5 (0.1)**	86.6 (0.2)	**87.5 (0.1)**
Weather						
Snow	76.0	80.4 (0.0)	82.2 (0.0)	**82.8 (0.1)**	82.3 (0.2)	**82.8 (0.1)**
Frost	64.8	80.0 (0.0)	81.1 (0.0)	**82.4 (0.3)**	81.9 (0.2)	**82.4 (0.3)**
Fog	73.9	83.9 (0.0)	84.9 (0.0)	**85.5 (0.2)**	84.1 (0.2)	**85.5 (0.2)**
Brit.	90.8	90.2 (0.0)	91.0 (0.0)	**91.1 (0.0)**	91.0 (0.1)	**91.1 (0.0)**
Digital						
Elastic	74.5	74.8 (0.0)	**76.3 (0.0)**	76.3 (0.1)	76.2 (0.2)	**76.3 (0.1)**
Pixelate	45.0	76.9 (0.0)	79.3 (0.0)	**80.1 (0.1)**	79.9 (0.2)	**80.1 (0.1)**
Avg.	55.2	76.0 (0.0)	77.9 (0.0)	**78.5 (0.2)**	78.2 (0.2)	78.5 (0.2)

Table 4: Accuracy (%) with standard deviation across different corruption types and adaptation methods using ResNet-18 on CIFAR-10-C (severity 5). Accuracy is computed on 5000 samples, processed in batches of 100 during adaptation. The highest accuracy per corruption type is bold, and the second-highest is underlined. The remaining five corruption types are used to train our method.

Corruption	No Adapt	SAR	TENT	DEYO	ROID	CoTTA	EATA	FOA	Surgeon	SPA	Our (NC)	Our (Ent.)
Noise												
Shot	56.0 (0.3)	59.2 (0.4)	58.3 (0.3)	59.4 (0.4)	61.4 (0.4)	58.8 (2.1)	59.2 (0.3)	59.1 (0.3)	60.3 (0.3)	**62.4 (0.4)**	**62.4 (0.4)**	**62.4 (0.4)**
Blur												
Defocus	46.3 (0.4)	51.9 (0.6)	49.0 (0.7)	52.7 (0.4)	**56.8 (0.4)**	48.9 (2.6)	51.8 (0.6)	45.2 (0.5)	55.1 (0.5)	54.6 (1.0)	54.6 (1.0)	54.6 (1.0)
Motion	52.5 (0.6)	57.1 (0.7)	54.6 (0.7)	57.5 (0.7)	60.8 (0.6)	56.4 (0.7)	56.8 (0.8)	51.5 (0.5)	58.8 (0.7)	**63.8 (0.3)**	**63.8 (0.3)**	**63.8 (0.3)**
Zoom	43.8 (0.3)	50.0 (0.3)	46.7 (0.5)	50.3 (0.2)	55.4 (0.4)	47.9 (1.3)	49.4 (0.4)	43.4 (0.3)	51.8 (0.2)	**60.1 (0.3)**	60.1 (0.3)	**60.1 (0.3)**
Weather												
Snow	61.8 (0.6)	62.7 (0.6)	62.2 (0.6)	63.2 (0.7)	66.1 (0.9)	65.6 (0.5)	62.9 (0.6)	64.4 (1.0)	64.2 (0.6)	**69.9 (0.5)**	**69.9 (0.5)**	**69.9 (0.5)**
Frost	62.3 (0.8)	58.2 (1.0)	58.5 (0.8)	58.5 (0.9)	62.8 (0.5)	64.8 (0.7)	58.5 (0.7)	61.3 (2.4)	60.9 (0.8)	**69.0 (0.6)**	**69.0 (0.6)**	**69.0 (0.6)**
Fog	65.0 (0.6)	56.0 (15.0)	47.3 (1.4)	62.5 (1.1)	67.8 (0.6)	65.2 (2.5)	62.6 (0.7)	68.5 (0.5)	37.5 (15.6)	**70.4 (0.6)**	67.8 (0.6)	67.8 (0.6)
Digital												
Elastic	45.3 (0.2)	50.4 (0.5)	47.4 (0.4)	51.4 (0.3)	58.3 (0.5)	50.2 (0.7)	50.3 (0.3)	47.6 (0.6)	52.8 (0.6)	**66.3 (0.8)**	**66.3 (0.8)**	**66.3 (0.8)**
Pixelate	66.4 (0.3)	67.8 (0.3)	67.1 (0.4)	68.4 (0.4)	70.1 (0.3)	69.0 (1.1)	67.9 (0.4)	65.8 (0.2)	69.8 (0.3)	**74.7 (0.2)**	72.0 (2.4)	70.1 (0.3)
JPEG	66.3 (0.3)	67.9 (0.4)	67.1 (0.4)	68.1 (0.4)	68.8 (0.3)	68.3 (0.4)	67.8 (0.4)	66.5 (0.7)	68.6 (0.3)	**71.6 (0.3)**	69.3 (1.2)	**71.6 (0.3)**
Avg.	56.6 (0.4)	58.1 (2.0)	55.8 (0.6)	59.2 (0.5)	62.8 (0.5)	59.5 (1.3)	58.7 (0.5)	57.3 (0.6)	58.0 (2.0)	**66.3 (0.5)**	65.5 (0.8)	65.5 (0.5)

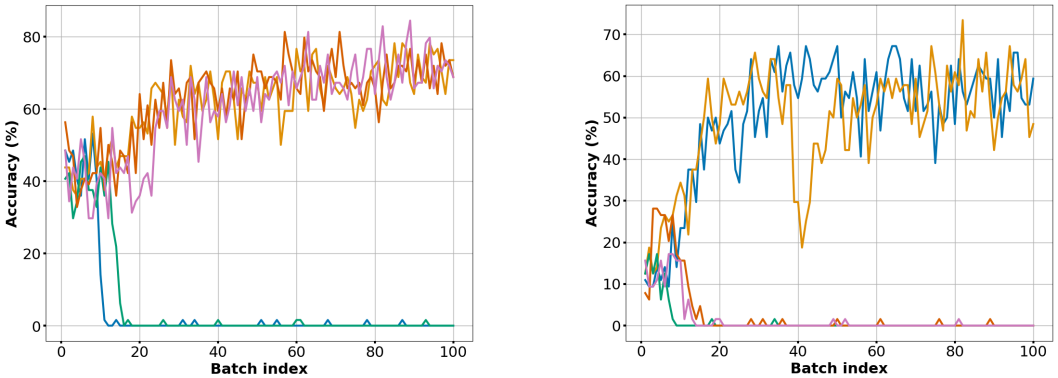
Table 5: Accuracy (%) with standard deviation across different corruption types and adaptation methods with ViT-Base on ImageNet-C (severity 5). Accuracy is calculated on the 6400 samples used to adapt. The highest accuracy per corruption type is bold, and the second-highest is underlined. The remaining five corruption types were used to train our method. (1)

Corruption	No Adapt	SAR	TENT	DEYO	ROID	CoTTA	EATA	FOA	Surgeon	SPA	Our (NC)	Our (Ent.)
Noise												
Gaussian	55.3 (0.4)	58.0 (0.2)	57.5 (0.2)	58.2 (0.3)	59.9 (0.4)	54.3 (7.7)	58.0 (0.3)	57.3 (0.5)	59.1 (0.3)	**61.0 (0.5)**	**61.0 (0.5)**	**61.0 (0.5)**
Shot	56.0 (0.3)	59.2 (0.4)	58.3 (0.3)	59.4 (0.4)	61.4 (0.4)	58.8 (2.1)	59.2 (0.3)	59.0 (0.3)	60.3 (0.3)	**62.4 (0.4)**	**62.4 (0.4)**	**62.4 (0.4)**
Impulse	56.0 (0.1)	59.4 (0.2)	58.6 (0.2)	59.5 (0.1)	61.2 (0.4)	53.4 (8.9)	59.3 (0.3)	58.0 (0.3)	60.6 (0.2)	**62.5 (0.5)**	**62.5 (0.5)**	**62.5 (0.5)**
Blur												
Glass	34.4 (0.4)	45.1 (0.8)	37.7 (0.7)	45.8 (0.6)	52.3 (0.3)	38.5 (0.5)	44.1 (0.4)	33.8 (0.6)	45.7 (0.5)	**55.0 (0.2)**	**55.0 (0.2)**	**55.0 (0.2)**
Weather												
Snow	61.8 (0.6)	62.7 (0.6)	62.2 (0.6)	63.2 (0.7)	66.1 (0.9)	65.6 (0.5)	62.9 (0.6)	64.4 (1.0)	64.2 (0.6)	**69.9 (0.5)**	**69.9 (0.5)**	**69.9 (0.5)**
Fog	65.0 (0.6)	56.0 (15.0)	47.3 (1.4)	62.5 (1.1)	67.8 (0.6)	65.2 (2.5)	62.6 (0.7)	68.5 (0.5)	37.5 (15.6)	**70.4 (0.6)**	67.8 (0.6)	67.8 (0.6)
Brightness	77.2 (0.4)	77.5 (0.5)	77.2 (0.4)	77.7 (0.4)	78.3 (0.4)	77.7 (0.4)	77.7 (0.4)	76.9 (0.4)	78.4 (0.4)	**79.8 (0.4)**	**79.8 (0.4)**	**79.8 (0.4)**
Digital												
Contrast	31.9 (0.4)	18.1 (18.6)	49.4 (0.5)	45.5 (24.7)	**62.9 (0.6)**	35.0 (10.7)	54.4 (0.4)	47.4 (0.9)	53.5 (0.5)	57.6 (6.9)	**62.9 (0.6)**	**62.9 (0.6)**
Pixelate	66.4 (0.3)	67.8 (0.3)	67.1 (0.4)	68.4 (0.4)	70.1 (0.3)	69.0 (1.1)	67.9 (0.4)	65.8 (0.2)	69.8 (0.3)	**74.7 (0.2)**	**74.7 (0.2)**	**74.7 (0.2)**
JPEG	66.3 (0.3)	67.9 (0.4)	67.1 (0.4)	68.1 (0.4)	68.8 (0.3)	68.3 (0.4)	67.8 (0.4)	66.5 (0.7)	68.6 (0.3)	**71.6 (0.3)**	**71.6 (0.3)**	**71.6 (0.3)**
Avg.	57.0 (0.4)	57.2 (3.7)	58.2 (0.5)	60.8 (2.9)	64.9 (0.5)	58.6 (3.5)	61.4 (0.4)	59.8 (0.6)	59.8 (1.9)	66.5 (1.1)	**66.8 (0.4)**	**66.8 (0.4)**

Table 6: Accuracy (%) with standard deviation across different corruption types and adaptation methods with ViT-Base on ImageNet-C (severity 5). Accuracy is calculated on the 6400 samples used to adapt. The highest accuracy per corruption type is bold, and the second-highest is underlined. The remaining five corruption types were used to train our method. (2)

Corruption	No Adapt	SAR	TENT	DEYO	ROID	CoTTA	EATA	FOA	Surgeon	SPA	Our (NC)	Our (Entr.)
Noise												
Gaussian	55.3 (0.4)	58.0 (0.2)	57.5 (0.2)	58.2 (0.3)	59.9 (0.4)	54.3 (7.7)	58.0 (0.3)	57.3 (0.5)	59.1 (0.3)	**61.0 (0.5)**	58.0 (0.3)	**61.0 (0.5)**
Impulse	56.0 (0.1)	59.4 (0.2)	58.6 (0.2)	59.5 (0.1)	61.2 (0.4)	53.4 (8.9)	59.3 (0.3)	58.0 (0.3)	60.6 (0.2)	**62.5 (0.5)**	60.0 (1.9)	**62.5 (0.5)**
Blur												
Defocus	46.3 (0.4)	51.9 (0.6)	49.0 (0.7)	52.7 (0.4)	**56.8 (0.4)**	48.9 (2.6)	51.8 (0.6)	45.2 (0.5)	55.1 (0.5)	54.6 (1.0)	50.7 (5.3)	54.6 (1.0)
Glass	34.4 (0.4)	45.1 (0.8)	37.7 (0.7)	45.8 (0.6)	52.3 (0.3)	38.5 (0.5)	44.1 (0.4)	33.8 (0.6)	45.7 (0.5)	**55.0 (0.2)**	**55.0 (0.2)**	**55.0 (0.2)**
Motion	52.5 (0.6)	57.1 (0.7)	54.6 (0.7)	57.5 (0.7)	60.8 (0.6)	56.4 (0.7)	56.8 (0.8)	51.5 (0.5)	58.8 (0.7)	**63.8 (0.3)**	56.8 (0.8)	**63.8 (0.3)**
Zoom	43.8 (0.3)	50.0 (0.3)	46.7 (0.5)	50.3 (0.2)	55.4 (0.4)	47.9 (1.3)	49.4 (0.4)	43.4 (0.3)	51.8 (0.2)	**60.1 (0.3)**	49.4 (0.4)	**60.1 (0.3)**
Weather												
Frost	62.3 (0.8)	58.2 (1.0)	58.5 (0.8)	58.5 (0.9)	62.8 (0.5)	64.8 (0.7)	58.5 (0.7)	61.3 (2.4)	60.9 (0.8)	**69.0 (0.6)**	64.7 (5.3)	**69.0 (0.6)**
Brightness	77.2 (0.4)	77.5 (0.5)	77.2 (0.4)	77.7 (0.4)	78.3 (0.4)	77.7 (0.4)	77.7 (0.4)	76.9 (0.4)	78.4 (0.4)	**79.8 (0.4)**	**79.8 (0.4)**	**79.8 (0.4)**
Digital												
Contrast	31.9 (0.4)	18.1 (18.6)	49.4 (0.5)	45.5 (24.7)	**62.9 (0.6)**	35.0 (10.7)	54.4 (0.4)	47.4 (0.9)	53.5 (0.5)	57.6 (6.9)	57.6 (6.9)	57.6 (6.9)
Elastic	45.3 (0.2)	50.4 (0.5)	47.4 (0.4)	51.4 (0.3)	58.3 (0.5)	50.2 (0.7)	50.3 (0.3)	47.6 (0.6)	52.8 (0.6)	**66.3 (0.8)**	**66.3 (0.8)**	**66.3 (0.8)**
Avg.	50.5 (0.4)	52.6 (2.3)	53.7 (0.5)	55.7 (2.9)	60.9 (0.5)	52.7 (3.4)	56.0 (0.5)	52.3 (0.7)	57.7 (0.5)	**63.0 (1.1)**	59.8 (2.2)	**63.0 (1.1)**

Table 7: Accuracy (%) with standard deviation across different corruption types and adaptation methods with ViT-Base on ImageNet-C (severity 5). Accuracy is calculated on the 6400 samples used to adapt. The highest accuracy per corruption type is bold, and the second-highest is underlined. The remaining five corruption types were used to train our method. (3)



(a) Fog (b) Contrast

Figure 5: Accuracy per batch for a model adapted using the SPA method on (a) Fog and (b) Contrast (ImageNet-C) domains. Results are shown for five different batch orders, considering only the first 100 batches. This results are obtained if we do not apply pre-processing to the ImageNet-C images (e.g. without normalisation).