Test-Time Training with Masked Autoencoders - Topic J

Marguerite Petit Talamon, Gabin Agbalé
{marguerite.petit-talamon, gabin.agbale}@dauphine.eu

Abstract

Test-Time Training is an approach that aims at better generalization by using the hint on the data distribution given by test data. It allows the model to adapt its parameters during Test-Time in a self-supervised way, such that it keeps learning better feature representations even during testing. The goal of this project is to explore and analyze this Test Time Training (TTT) technique.

1. Introduction

1.1. Problem of Distribution Shift

In supervised learning, it is commonly assumed that the test and training sets share the same distribution. However, in real-world scenarios, this assumption often does not hold true. A typical example of distributional shift between datasets is the presence of varying noise distributions.

Figure 1. **Examples of blur corrupted images from ImageNet-C dataset [4].** First row of images are corrupted in level 1. The second row level 5. From left to right, the noises are *defocus blur*, *motion blur*, *glass blur* and *zoom blur*.

In an image classification task, a crucial step is to derive a vector representation of the image, known as a latent space representation, which is then used for classification. The model is trained to effectively represent the input (in this case, an image) as a vector. However, when noise is introduced or when images outside the training set are encountered, the quality of this input representation can significantly deteriorate, leading to incorrect classifications.

This phenomenon is illustrated in Figure 4, where we used Principal Component Analysis (PCA) to project the latent representations of two classes affected by three different types of noise. The PCA, a dimensionality reduction technique that emphasizes variance, reveals a striking observation: in the latent space, the noise in the images becomes a prominent feature. The segmentation based on noise differences is quite clear, whereas the differentiation based on class differences is almost negligible. This poses a significant challenge, especially when the primary objective is to classify images accurately based on their class.

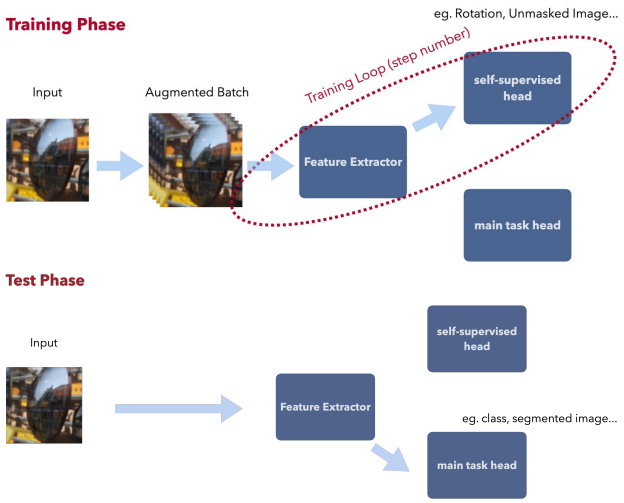
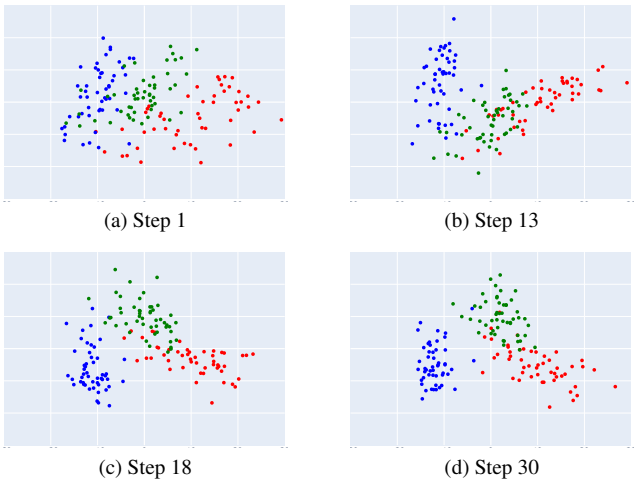


Figure 2. **Simplified MAE TTT model**

1.2. Test-Time Training

Test-Time Training is a test method that allows for the model to keep learning features on test samples. It consists in associating a *self-supervised task* (in our case a *Mask Auto Encoder* [3]) to the *main task* using a *Y-shape model*. Let $\theta := (\theta_e, \theta_d)$ be the respective parameters of the encoder and the decoder obtained after pre-training. For every sample x in the test set, we train both the encoder and decoder a given number of steps, and obtain parameters θ_*. Parameters θ_e* are then used to classify x with θ_m the parameters of the *classification head* which remain fixed. We

discard θ_* back to θ before moving to the next test sample.



(a) Step 1 (b) Step 13

(c) Step 18 (d) Step 30

Figure 3. **Evolution of the latent representation from MAE of images from three different categories**. We visualize the evolution of latent representations for three image categories: 'Indigo Bunting' (red), 'Earth Star' (blue), and 'Axolotl' (green). The data, reduced using PCA, was analyzed over 30 TTT steps with a batch size of one, showing increasingly distinct clusters for each category)

1.3. Experimentations

To conduct this experiment and subsequent ones detailed in this report, our approach was grounded in the methodologies and code from the *Test Time Training with Masked AutoEncoder* [2] and *ImageNet-C* dataset [4]. We used pre-trained weights for both the Masked Autoencoder (MAE) and the model head. This decision was made to leverage the advanced capabilities and learned features of the MAE, ensuring a robust starting point for our experiments. The pre-trained weights provide a well-established foundation, which is particularly beneficial in handling complex tasks like image classification and dealing with distribution shifts in the data. This approach allowed us to focus on the specific aspects of Test Time Training (TTT) and its effectiveness under varying conditions, such as different noise distributions and batch sizes, as highlighted in our experiments.

2. Application of TTT-MAE *GA & MPT*

The first objective of our study was to replicate the results using the TTT-MAE approach on the ImageNet-C dataset, as presented in the paper [2]. ImageNet-C consists of 64x64 images across 1000 classes, subjected to 15 types of corruptions at 5 different severity levels. Due to the vast size of the dataset, containing 3,750,000 images, our analysis was conducted on a subset. This subset comprised 5000 images, sampled from 50,000, across five different corruptions at the highest severity level, level 5.

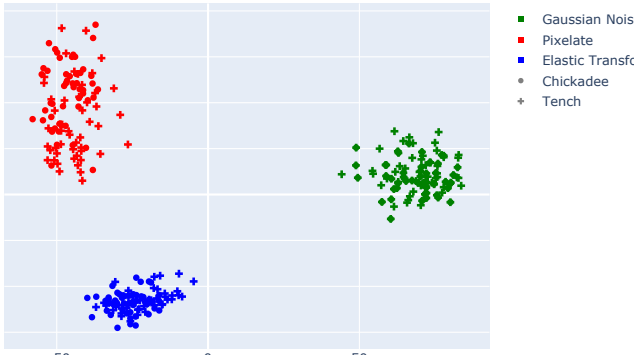


Figure 4. **Problem of distribution shift**. Latent representations from Masked AutoEncoder pre trained of Tench and Chickadee classes with three different noise applied.

Following the methodology in the TTT paper, we first evaluated the model's accuracy across different types of corruptions, as depicted in Figure 1. However, our experimental settings differed from those in the TTT paper. Unlike the original study, which assessed accuracy after 10 steps of Test Time Training (TTT) using a batch size of 128 and an SGD optimizer, our evaluation was conducted after the same number of TTT steps but with a batch size of 8 and an SGD optimizer. This choice was influenced by computing cost considerations. It's important to note that one step of TTT for a single image with a batch size of 128 entails training on 128 images and then evaluating the target image.

In order to demonstrate the impact of batch size in TTT, we compared the performance for two different batch sizes on the same corruption type and level. As illustrated in Figure 2, the accuracy on the 'zoom blur' corruption at level 5 improved by 8.9% after 10 steps of TTT with a batch size of 8. In contrast, the accuracy increased by only 0.99% with a batch size of 1.

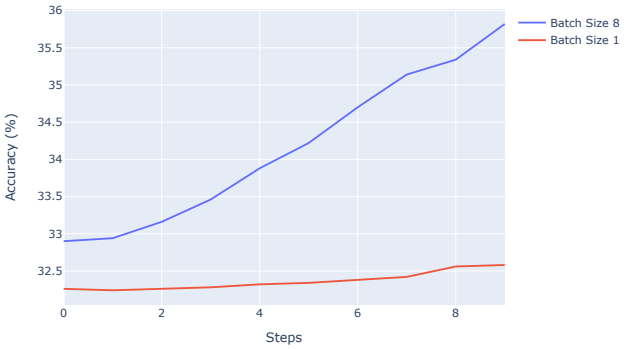


Figure 5. **AdamW vs SGD during TTT**. We evaluated the accuracy during 30 steps of Test Time Training with 5000 *zoom blur* corrupted images with a batch size of 1.

In the next experiment, we compared the use of AdamW and SGD optimizers on Test Time Training. As indicated in the article [2] and Figure 6, AdamW requires early stopping. AdamW is a variant of the Adam algorithm, it stands out by separating weight decay from optimization, potentially improving training and generalization. AdamW converges faster than SGD, which can lead to overfitting if not closely monitored. This is due to AdamW's adaptive learning rate, which makes updates more aggressive than SGD's steadier approach. However, AdamW is sensitive to hyperparameters like learning rate and weight decay, necessitating careful adjustment. Due to computational limits, we could not extensively test AdamW's hyperparameters which would be an interesting experiment to conduct.

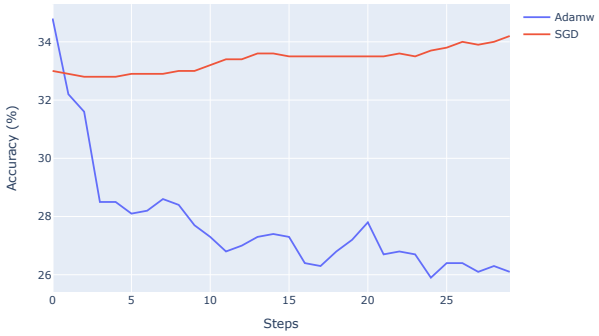


Figure 6. **AdamW vs SGD during TTT**. We evaluated the accuracy during 30 steps of Test Time Training with 5000 *zoom blur* corrupted images with a batch size of 1.

Table 1. Accuracy (%) on ImageNet-C, level 5, after 10 steps of TTT for selected corruptions.

	gauss	mtn	zoom	contr	pix
Baseline	17.4	32.2	32.2	6.4	49.7
TTT-MAE [128]	25.1	39.5	42.9	9.2	59.9
TTT-MAE [8]	17.3	33.5	34.2	7	51.4

3. Online-TTT

The Online-TTT is an alternative to the Standard version of TTT, that was introduced in [5]. This method has shown improved accuracy in the case of *Rotation-Test Time Training*, we propose an equivalent study for *MAE-TTT*. Online-TTT differs from Standard version by retaining the parameters θ_* obtained after classification on test sample x_t, instead of discarding them. Thus, the parameters θ_{t-1} obtained after applying Test-Time Training on x_{t-1} are kept as a starting point for Test-Time Training on x_t. This allows θ_t to also leverage the distributions observed on $x_0, x_1, ..., x_{t-1}$. In the case of the Online version, the number of *TTT-steps* is fixed as one. We consider 3 different assumptions on the data settings, on which we experiment with Online-TTT. We will show that Online-TTT improves performance in nearly every case, as shown in 7.

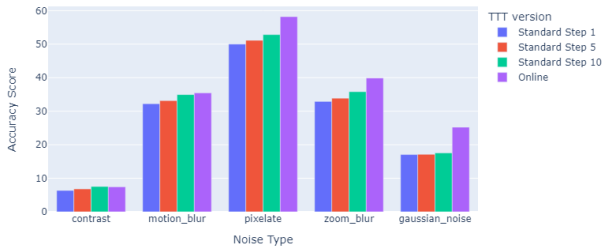


Figure 7. **Comparison of the accuracy score of Online TTT and at different steps of Standard TTT** This experiment has been realised on 5000 images for 5 different type of noise.

3.1. Online-TTT on separate Noises

We first perform Online-TTT on noises separately, for 5000 test iterations. We also evaluated the model every 100 iterations on an independant evaluation set. In the case of *zoom blur* noise, we obtain an accuracy of 39.9% on the test set (as compared to the 34.1% accuracy obtain with Standard-TTT in the same setting) and an evaluation accuracy of 40.8% (see 8). Thus it seems that Online-TTT not only improves accuracy over the test samples but also ensures better generalization over unseen validation data. One can notice the intense fluctuations in the moving average accuracy of Online-TTT. Our study as led to deduce that this was not a consequence of some classes having a bad accuracy. Indeed, the moving average is supposed to mitigate the direct impact of classes with low accuracy score. Moreover, experiments with Online-TTT involved a shuffled class order (contrary to Standard-TTT where classes are evaluated consecutively).

3.2. Shuffled Noises vs Consecutive Noises

In order to test the added-value of performing Online-TTT on separate noise types, we applied the method to a dataset containing images from random types of noise. We restricted our study to the case of *blur noise*. The accuracy obtained on the shuffled *blur noise* dataset is 31.3%, which is lower than the average accuracy of 33% obtained when applying Online-TTT on noises separately on an equivalent number of iterations (5000). We compare it to the

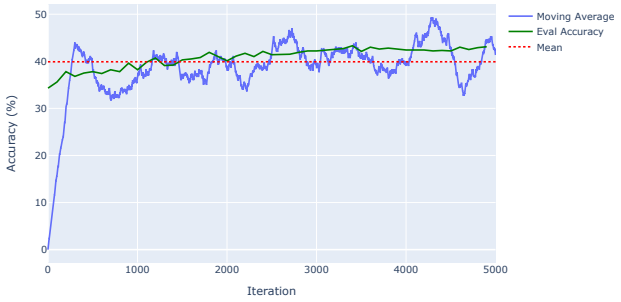


Figure 8. **Evolution of the accuracy during Online-TTT on a unique noise**. We evaluated the accuracy of On TTT on 5000 *zoom blur* corrupted images with a batch size of 8.

case where we apply Online-TTT consecutively on different types of *blur noise* (as shown in 10, where we obtain a lower test accuracy of 32.2% and evaluation accuracy of 31.9%.

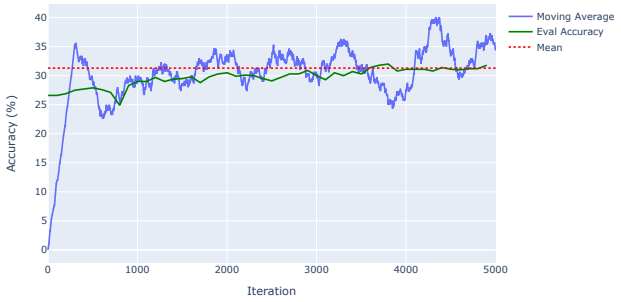


Figure 9. **Evolution of the accuracy during Online TTT on shuffled noises.** We evaluated the accuracy of Online TTT on 5000 *random blur* corrupted images with a batch size of 8. *Blur* noises contain *defocus blur*, *zoom blur*, *glass blur* and *motion blur*.

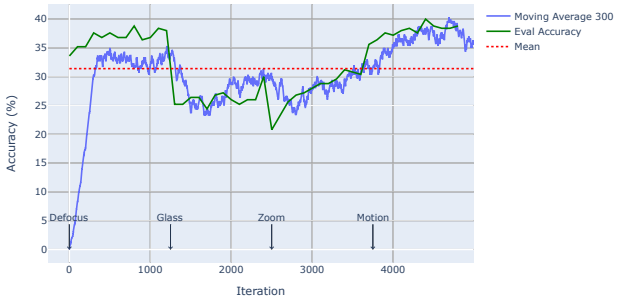


Figure 10. **Evolution of the accuracy during Online TTT on classified *blur* noises.** We evaluated the accuracy of Online TTT on 5000 *blur* corrupted images with a batch size of 8. The first 1250 iterations are *defocus blur* corrupted images, the next 1250 are *glass blur* corrupted images, etc.

3.3. Gradual Distribution Shift

Instead of assuming that all test data $x_0, x_1, ..., x_t$ are sampled from the same distribution Q (which is true if $x_0, x_1, ..., x_t$ are iid), we assume that the test data distribution shifts gradually. In other words, we have $x_t \sim Q_t$ with Q_t closer to Q_{t-1} than to Q_{t-2} for all t. To illustrate that with our Dataset ImageNet-C, we performed Online-TTT on a given noise, first on test samples with noise intensity σ_1, then σ_2, up until σ_5, with $\sigma_1 <, ..., < \sigma_5$. This was applied to *zoom blur* noise as illustrated in 11, where we obtain an accuracy on 5000 test samples of 55.3% and an evaluation accuracy of 55.1%. Notably, when applying Online-TTT under the context of Gradual Distribution Shift, we observed superior outcomes compared to scenarios involving shuffled noise levels, as depicted in Figure 12. In this case, the test accuracy stood at 53.8%, while the evaluation accuracy reached 54.4%. One can remark that this does not perfectly match the *Gradual Distribution Shift* assumption, in which the distribution shifts along every data point. To reproduce this setting, we could have drawn interpolations between every 5 noise levels and gradually increase the weight of noise level 5.

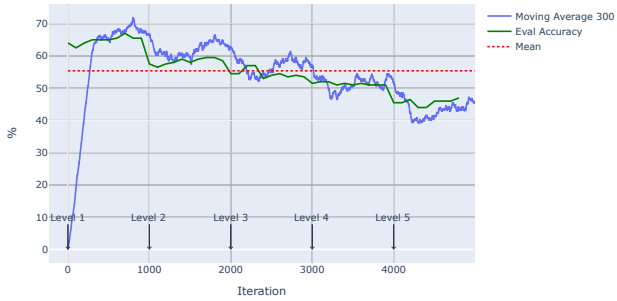


Figure 11. **Evolution of the accuracy during Online TTT on classified *zoom blur* levels.** We evaluated the accuracy of Online TTT on 5000 *blur* corrupted images with a batch size of 8. The first 1000 iterations are level 1 *zoom blur* corrupted images, the next 1000 are level 2 corrupted images, etc.

4. Wrongly classified Images *GA*

In our study, we noticed a significant variation in accuracy across different classes. Initially, we hypothesized that this inconsistency might be related to the size of the objects within the images. To explore this possibility, we turned to the ImageNet1k dataset, which includes uncorrupted images from ImageNetC. Our approach involved extracting the bounding boxes of objects in these images to gain insight into their size relative to the overall image.

We calculated a ratio for each object by dividing the area of the object's bounding box by the total image size, thereby obtaining a series of ratios for each class. Our aim

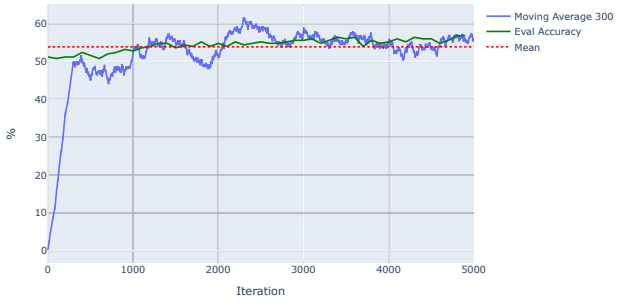


Figure 12. **Evolution of the accuracy during Online TTT on random level of *blur zoom* noise.** We evaluated the accuracy of Online TTT on 5000 *zoom blur* corrupted images with different level of corruption and with a batch size of 8.

.

was to determine whether there was a correlation between these size ratios and the classification accuracy for each class.

To test this hypothesis, we conducted an analysis focusing on the variance and mean of these ratios within each class. We sought to identify any significant patterns or relationships that could link these ratio metrics to the corresponding accuracy levels for each class. However, the results of this analysis were not conclusive, indicating that the object size, as represented by our calculated ratios, did not have a clear or direct impact on the accuracy variation among different classes.

Since we were not able to validate this hypothesis, we searched for categories where the accuracy did not improved or decreased from baseline to Test Time Training. We did not arrived to any relevant findings.

5. Conclusion

This approach allows a better average accuracy on corrupted images, an interesting idea would be to test the technique on random images (not necessarily corrupted). Our study as shown that the *Online* version of *Test Time Training* ensures better accuracy than the *Standard* version with a lower computation cost, even though its accuracy varies a lot during test time. One could think of a way to reduce these variations and ensure consistent results. Moreover, the computing time of TTT is longer than any other testing method (*number of of steps * batch size* longer). This would be an interesting subject to search about. Also, this TTT method exists with Masked AutoEncoder, we could evaluate the model with other self supervised task.

References

[1] Please find code of the project here. Our code is an extension of this code.

[2] Y. Gandelsman, Y. Sun, X. Chen, and A. A. Efros. "Test-time training with masked autoencoders." In *Advances in Neural Information Processing Systems*, 2022.

[3] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. "Masked Autoencoders Are Scalable Vision Learners," 2021. 2, 3

[4] D. Hendrycks and T. G. Dietterich. "Benchmarking neural network robustness to common corruptions and perturbations." *CoRR*, abs/1903.12261, 2019. 1

[5] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt. "Test-time training with self-supervision for generalization under distribution shifts." *International Conference on Machine Learning*, 2020. 1, 2