Video Test-Time Adaptation for Action Recognition

Wei Lin[†*1,2] Muhammad Jehanzeb Mirza[*1,3] Mateusz Kozinski[1] Horst Possegger[1]
Hilde Kuehne[4,5] Horst Bischof[1,3]

[1]Institute for Computer Graphics and Vision, Graz University of Technology, Austria
[2]Christian Doppler Laboratory for Semantic 3D Computer Vision
[3]Christian Doppler Laboratory for Embedded Machine Learning
[4]Goethe University Frankfurt, Germany
[5]MIT-IBM Watson AI Lab

Abstract

Although action recognition systems can achieve top performance when evaluated on in-distribution test points, they are vulnerable to unanticipated distribution shifts in test data. However, test-time adaptation of video action recognition models against common distribution shifts has so far not been demonstrated. We propose to address this problem with an approach tailored to spatio-temporal models that is capable of adaptation on a single video sample at a step. It consists in a feature distribution alignment technique that aligns online estimates of test set statistics towards the training statistics. We further enforce prediction consistency over temporally augmented views of the same test video sample. Evaluations on three benchmark action recognition datasets show that our proposed technique is architecture-agnostic and able to significantly boost the performance on both, the state of the art convolutional architecture TANet and the Video Swin Transformer. Our proposed method demonstrates a substantial performance gain over existing test-time adaptation approaches in both evaluations of a single distribution shift and the challenging case of random distribution shifts. Code will be available at `https://github.com/wlin-at/ViTTA`.

1. Introduction

State-of-the-art neural architectures [8,40,46,48–50] are very effective in action recognition, but recent work shows they are not robust to shifts in the distribution of the test data [33,51]. Unfortunately, in practical scenarios, such distribution shifts are very difficult to avoid, or account for. For example, cameras used for recognizing motorized or pedestrian traffic events may register rare weather conditions, like a hailstorm, and sports action recognition systems can be affected by perturbations generated by spectators at sports arenas, such as the smoke of flares. Shifts in the data distribution can also result from inconspicuous changes in the video processing setup, for instance, a change of the algorithm used to compress the video feed.

In image classification, distribution shift can be mitigated by Test-Time-Adaptation (TTA) [16, 19, 21, 23, 28, 34, 38, 42], which uses the unlabeled test data to adapt the model to the change in data distribution. However, methods developed for image classification are not well suited for action recognition. Most action recognition applications require running memory- and computation-hungry temporal models online, with minimal delay, and under tight hardware constraints. Moreover, videos are more vulnerable to distribution shifts than images [2,33,51]. Some examples of such distribution shifts are given in Fig. 1. Due to limited exposure times, video frames are likely to feature higher variations of noise level, provoked by illumination changes. They are more affected by motion blur, which varies with the speed of motion observed in the scene. They also feature stronger compression artifacts, which change with the compression ratio, often dynamically adjusted to the available bandwidth. Our experiments show that existing TTA algorithms, developed for image classification, do not cope with these challenges well, yielding marginal improvement over networks used on corrupted data without any adaptation.

Our goal is to propose an effective method for online test-time-adaptation of action recognition models. Operating online and with small latency can require drastically constraining the batch size, especially when the hardware resources are limited and the employed model is large. We therefore focus on the scenario in which test samples are

∗ Equally contributing authors.
† Correspondence: wei.lin@icg.tugraz.at

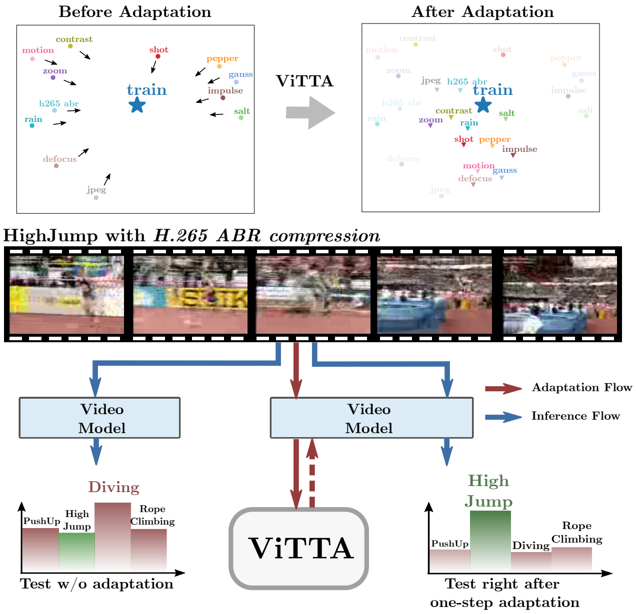

Figure 1. Our architecture-agnostic ViTTA enables action recognition models to overcome severe video corruptions. In a fully online manner, *i.e.* processing each test video only once, we perform test-time adaptation. In particular, we align the statistics of the (corrupted) test data towards the (clean) training data such that they are better aligned (top: t-SNE [41] of mean features from final feature extractor layer for clean training data and for 12 differently corrupted test datasets). This results in a significant performance improvement for action recognition.

processed individually, one at a time. To ease the integration of our method in existing systems, we require it to be capable of adapting pretrained networks, both convolutional and transformer-based, without the need to retrain them.

In the work, we propose the first video test-time adaptation approach - ViTTA. To address the above requirements, we turn to feature alignment [19, 28, 34, 37, 53], a common TTA method that aligns distributions of features computed for test and training data by minimizing discrepancy between their statistics. Feature alignment does not require any modifications to the training procedure and is architecture-agnostic. However, existing feature alignment methods are not well suited for online adaptation, because they require relatively large test batches to accurately estimate the statistics. We address this by employing the exponential moving average to estimate test feature statistics online. This enables us to perform the alignment by processing one video sample at a time, at a low computational and memory cost. Additionally, we show that even though the temporal dimension of video data poses challenges, it also has a silver lining. We leverage this by creating augmented views of the input videos via temporally resampling frames in the video. This has two benefits: First, multiple augmented views lead to more accurate statistics of the overall video content. Second, it allows us to enforce prediction consistency across the views, making the adaptation more effective.

Our extensive evaluations on three most popular action recognition benchmarks demonstrate that ViTTA boosts the performance of both TANet [27], the state-of-the-art convolutional architecture, and the Video Swin Transformer [26], and outperforms the existing TTA methods proposed for image data by a significant margin. ViTTA performs favorably in both, evaluations of single distribution shift and the challenging case of random distribution shift.

ViTTA also has a high practical valor. It is fully online and applicable in use cases requiring minimal delays. It does not require collection or storage of test video dataset, which is significant in terms of data privacy protection, especially in processing confidential user videos. ViTTA can be seamlessly incorporated in systems already in operation, as it has no requirement of re-training existing networks. Therefore it can harness state-of-the-art video architectures.

Our contributions can be summarized as follows:

- We benchmark existing TTA methods in online adaptation of action recognition models to distribution shifts in test data, on three most popular action recognition datasets, UCF101 [35], Something-something v2 [13] and Kinetics 400 [18]

- We adapt the feature alignment approach to online action recognition, generating a substantial performance gain over existing techniques.

- We propose a novel, video-specific adaptation technique (ViTTA) that enforces consistency of predictions for temporally re-sampled frame sequences and show that it contributes to adaptation efficacy.

2. Related Work

Action Recognition is addressed mainly with CNN-based and transformer-based architectures. CNN-based architectures typically use 3D convolutions, such as C3D [39], I3D [5], Slowfast [9] and X3D [8]. Recent works also deploy 2D convolution with temporal modules [20, 22, 25, 27, 32, 43, 44] to reduce the computational overhead. TEINet [25] learns the temporal features by decoupling the modeling of channel correlation and temporal interaction for efficient temporal modeling. TANet [27] is a 2D CNN with integrated temporal adaptive modules which generates temporal kernels from its own feature maps, achieving state-of-the-art performance among convolutional competitors. Transformer-based models have also been applied for video recognition [1, 3, 7, 26, 30, 31, 54].

ViViT [1] adds several temporal transformer encoders on the top of spatial encoders. Video swin transformer [26] uses spatio-temporal local windows to compute the self-attention. In this work, we evaluate our adaptation method on TANet and Video Swin Transformer.

Robustness of Video Models for action recognition against common corruptions has recently been analyzed [33, 51]. Yi *et al*. [51] and Schiappa *et al*. [33] benchmark robustness of common convolutional- and transformer-based spatio-temporal architectures, against several corruptions in video acquisition and video processing. In this work, we perform evaluations on 12 corruptions proposed in these two benchmark works. These corruptions cover various types of noise and digital errors, blur effects of cameras, weather conditions, as well as quality degradation in image and video compression.

Test-Time Adaptation tackles the adaptation to unknown distribution shifts encountered at test-time in an unsupervised manner. It has recently gained increasing attention in the image domain [4,10,17,21,23,28,34,38,42,55]. These approaches can be divided into two distinct groups: The first group modifies the training procedure and employs a self-supervised auxiliary task to adapt to distribution shifts at test-time. Sun *et al*. [38] train the network jointly for self-supervised rotation prediction [12] and the main task of image classification. At test-time they adapt to the out-of-distribution test data by updating the encoder through the gradients obtained from the auxiliary task of rotation prediction on the test samples. TTT++ [23] propose self-supervised contrastive learning [6] as an auxiliary objective for adaptation and also aligns the source and target domain feature responses. Recently, Gandelsman *et al*. [10] use the self-supervised reconstruction task through masked autoencoders [14] for test-time adaptation.

The other group of methods, more closely related to our work, propose to adapt off-the-shelf pre-trained networks without altering the training. These methods typically employ post-hoc regularization. TENT [42] adapts a pre-trained network at test-time by minimizing the entropy from the output softmax distribution. Similarly, MEMO [55] proposes to adapt the network at test-time by minimizing the entropy of the marginal output distribution across augmentations. SHOT [21] also uses entropy minimization and adds information maximization regularization at test-time. On the other hand, some methods do not update the parameters for the network and instead propose gradient-free approaches for test-time adaptation. For example, LAME [4] only adapts the outputs of the network by using Laplacian regularization and guarantees convergence through a concave-convex optimization procedure. T3A [17] generates pseudo-prototypes from the test samples and replaces the classifier learned on the training set. DUA [28] and NORM [34] only update the statistics of the batch normalization layer for adaptation at test-time.

Despite the intensive development in the image domain, test-time adaptation on video action recognition models, to the best of our knowledge, has not been demonstrated so far. In this work, we propose a video-tailored adaptation method - ViTTA that adapts spatio-temporal models against common distribution shifts on video sequences. ViTTA consists in a feature distribution alignment technique that aligns training statistics with the online estimates of test set statistics. It is model-agnostic and can adapt to both convolutional and transformer-based network without the need to re-train them. We compare ViTTA to existing TTA approaches that do not require to alter the training of source model, and adapt to off-the-shelf pretrained-models.

3. Video Test-Time Adaptation (ViTTA)

We are given a multi-layer neural network ϕ trained in action recognition on a training set of video sequences S, and its optimal parameter vector $\hat{\theta}$, resulted from this training. At test time, the network is exposed to unlabeled videos from the test set T, that may be distributed differently than the data in S. Our goal is to adapt ϕ to this distribution shift to maximize its performance on the test videos. The pipeline of our method - ViTTA is shown in Fig. 2.

3.1. Feature distribution alignment

We perform the adaptation by aligning the distribution of feature maps computed for the training and test videos. Following recent work on TTA [28,34], to align the distributions, we equalize means and variances of the feature maps. We denote the feature map of the l-th layer of the network ϕ, computed for a video $\mathbf{x}$, by $\phi_l(\mathbf{x};\theta)$, where θ is the parameter vector used for the computation. The feature map is a tensor of size (c_l, t_l, h_l, w_l), where c_l denotes the number of channels in the l-th layer and t_l, h_l, and w_l are its temporal and spatial dimensions. We denote the spatio-temporal range of the feature map by $V = [1,t_l]\times[1,h_l]\times[1,w_l]$, and a c_l-element feature vector at voxel $v \in V$ by $\phi_l(\mathbf{x};\theta)[v]$. The mean vector of the l-th layer features for dataset D can be computed as the sample expectation

$$\mu_l(D;\theta) = \mathbb{E}_{\substack{\mathbf{x}\in D\\ v\in V}}\big[\phi_l(\mathbf{x};\theta)[v]\big], \tag{1}$$

and the vector of variances of the l-th layer features can be obtained as

$$\sigma_l^2(D;\theta) = \mathbb{E}_{\substack{\mathbf{x}\in D\\ v\in V}}\left[\big(\phi_l(\mathbf{x};\theta)[v] - \mu_l(D;\theta)\big)^2\right]. \tag{2}$$

To declutter the equations, we shorten the notation of the training statistics to $\hat{\mu}_l = \mu_l(S;\hat{\theta})$ and $\hat{\sigma}_l^2 = \sigma_l^2(S;\hat{\theta})$. In our experiments, we pre-compute them on the training data. When this data is no longer available, they can be substituted with statistics of another, unlabeled dataset, that

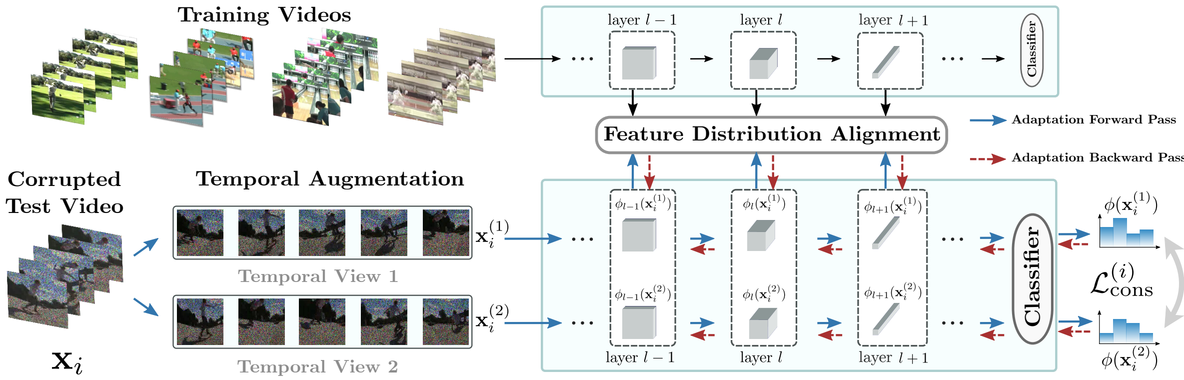

Figure 2. Pipeline of ViTTA. The online adaptation is applied on videos that are received sequentially and here we show the adaptation process of iteration i. We first compute the online estimates of the test statistics by 1) sampling two temporally augmented views from the test video, and computing the statistics on multi-layer features maps across the two views, 2) then performing exponential moving averages of statistics among iterations. Afterwards, we perform feature distribution alignment by minimizing the discrepancy between the pre-computed training statistics and the online estimates of test statistics. Furthermore, we enforce prediction consistency over temporally augmented views for performance boost.

is known to be generated from a similar distribution. In Sec. 4.4.2, we also show that for a network with batch normalization layers, running means and variances accumulated in these layers can be used instead of the statistics computed for the training set, with a small performance penalty.

Our overarching approach is to iteratively update the parameter vector θ in order to align the test statistics of selected layers to the statistics computed for the training data. This can be formalized as minimizing the alignment objective

$$\mathcal{L}_{\text{align}}(\theta) = \sum_{l \in L} |\mu_l(T;\theta) - \hat{\mu}_l| + |\sigma_l^2(T;\theta) - \hat{\sigma}_l^2| \quad (3)$$

with respect to the parameter vector θ, where L is the set of layers to be aligned, $|\cdot|$ denotes the vector l_1 norm, and we recall that T denotes the test set. Our approach of *training* the network to align the distributions is different in spirit from the TTA techniques based on feature alignment [28, 29, 34], which only adjust the running statistics accumulated during training in the normalization layers and therefore do not truly learn at test time. The fact that we update the entire parameter vector differentiates our method from the existing algorithms that only update parameters of affine transformation layers [19, 42], and gives it more flexibility during adaptation. Even though our method adjusts the full parameter vector, our experiments on continued adaptation (in Sec. 4.4.5), show that our methods adapts fast to periodic changes of distribution shift. When the distribution shift is removed from the stream of test data, the network quickly restores its original performance. We also found that the best performance was achieved by aligning the distributions of the features output by the last two out of four blocks for both TANet [27] and Video Swin Transformer [26]. We therefore set L to contain layers in these two blocks. The respective ablation study can be found in Sec. 4.4.3.

3.2. Online adaptation

Optimizing the objective in Eq. (3) requires iteratively estimating statistics of the test set. This is infeasible in an online video recognition system, typically required to process a stream of data with minimal delay. We therefore adapt the feature alignment approach to an online scenario. We assume the test data is revealed to the adaptation algorithm in a sequence of videos, denoted as $\mathbf{x}_i$, where i is the index of the test videos. We perform one adaptation step for each element of the sequence. Feature statistics computed on a single test sample do not represent feature distribution over the entire test set, so we cannot only rely on them when aligning the distributions. We therefore approximate test set statistics by exponential moving averages of statistics computed on consecutive test videos and use them for the alignment. We define the mean estimate in iteration i as

$$\mu_l{}^{(i)}(\theta) = \alpha \cdot \mu_l(\mathbf{x}_i;\theta) + (1-\alpha) \cdot \mu_l{}^{(i-1)}(\theta), \quad (4)$$

where $1-\alpha$ is the momentum that is set to a common choice of 0.9 ($\alpha = 0.1$). Similarly, we define the i-th variance estimate as

$$\sigma_l^2{}^{(i)}(\theta) = \alpha \cdot \sigma_l^2(\mathbf{x}_i;\theta) + (1-\alpha) \cdot \sigma_l^2{}^{(i-1)}(\theta). \quad (5)$$

To suit online adaptation, in the i-th alignment iteration, the objective in Eq. (3) is approximated by

$$\mathcal{L}_{\text{align}}^{(i)}(\theta) = \sum_{l \in L} |\mu_l{}^{(i)}(\theta) - \hat{\mu}_l| + |\sigma_l^2{}^{(i)}(\theta) - \hat{\sigma}_l^2|. \quad (6)$$

This approach simultaneously decreases the variance of the estimates and lets the network continuously adapt to changing distribution of test data.

3.3. Temporal augmentation

To further increase the efficacy of the method, we benefit from the temporal nature of the data and create M resampled views of the same video. We denote the temporally augmented views of the input video by $\mathbf{x}_i^{(m)}$, for $1 \leq m \leq M$. We compute the mean and variance vector of video $\mathbf{x}_i$ over the M views, to improve the accuracy of statistics on a single video:

$$\mu_l(\mathbf{x}_i;\theta) = \mathbb{E}_{\substack{m\in M\\ v\in V}}\left[\phi_l(\mathbf{x}_i^{(m)};\theta)[v]\right], \tag{7}$$

$$\sigma_l^2(\mathbf{x}_i;\theta) = \mathbb{E}_{\substack{m\in M\\ v\in V}}\left[\left(\phi_l(\mathbf{x}_i^{(m)};\theta)[v] - \mu_l(\mathbf{x}_i;\theta)\right)^2\right]. \tag{8}$$

We recall that $\mu_l(\mathbf{x}_i;\theta)$ and $\sigma_l^2(\mathbf{x}_i;\theta)$ are used in Eq. (4) and (5) for computing mean and variance estimate in iteration i.

Furthermore, we enforce consistency of the corresponding predictions among the M views. We establish a pseudo-label by averaging the class probabilities predicted by the network for the input views $y(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}\phi(\mathbf{x}_i^{(m)};\theta)$, and define the consistency objective in iteration i as

$$\mathcal{L}_{\text{cons}}^{(i)}(\theta) = \sum_{m=1}^{M}|\phi(\mathbf{x}_i^{(m)};\theta) - y(\mathbf{x})|. \tag{9}$$

In the i-th alignment iteration, we update the network parameter by following the gradient of

$$\min_{\theta}\mathcal{L}_{\text{align}}^{(i)}(\theta) + \lambda\cdot\mathcal{L}_{\text{cons}}^{(i)}(\theta), \tag{10}$$

where λ is the coefficient that we set to 0.1. In the ablation study, we show that setting $M = 2$ is enough to deliver a significant performance boost (Sec. 4.4.3), and that uniform equidistant resampling of the input videos yields the best results (Sec. 4.4.4).

4. Experiments

4.1. A Benchmark for Video Test-Time Adaptation

4.1.1 Video datasets.

We conduct experiments on three action recognition benchmark datasets: UCF101 [35], Something-something v2 (SSv2) [13] and Kinetics 400 (K400) [18]. The UCF101 dataset contains 13320 videos collected from YouTube with 101 action classes. We evaluate on split 1, which consists of 9537 training videos and 3783 validation videos. SSv2 is a large-scale action dataset with 168K training videos and 24K validation videos, comprised of 174 classes. K400 is the most popular benchmark for action recognition tasks, containing around 240K training videos and 20K validation videos in 400 classes.

4.1.2 Corruptions.

We evaluate on 12 corruption types proposed in [33,51] that benchmark robustness of spatio-temporal models. These 12 corruptions are: *Gaussian noise*, *pepper noise*, *salt noise*, *shot noise*, *zoom blur*, *impulse noise*, *defocus blur*, *motion blur*, *jpeg compression*, *contrast*, *rain*, *H.265 ABR compression*. They cover various types of noise and digital errors, blur effects of cameras, weather conditions, as well as quality degradation in image and video compression.

Following [33, 51], we evaluate on the validation sets of the three datasets. We use the implementation from these two benchmark papers for corruption generation. As the robustness analysis indicates approximate linear correlation between severity level and the performance drop, We evaluate on the corruptions of the most severe case at level 5.

4.2. Implementation Details

We evaluate our approach on two model architectures: TANet [27] based on ResNet50 [15], and Video Swin Transformer [26] based on Swin-B [24]. TANet is the state-of-the-art convolutional network for action recognition and Video Swin Transformer is adapted from Swin Transformer [24]. More model specifics are given in supplementary.

We perform distribution alignment of features from the normalization layers in the last two blocks. We set the batch size to 1 for evaluation on all datasets. We adapt to each video only once. Following common practice in online test-time adaptation [4, 38, 45], we perform inference right after adapting to a sample and report the accumulated accuracy for all samples. On TANet, we set the learning rate to $5e-5$ for UCF and $1e-5$ for SSv2 and K400. On Video Swin Transformer, learning rate is set to $1e-5$ for all datasets. For the temporal augmentation, we perform uniform equidistant sampling and random spatial cropping.

4.3. Comparison to State-of-the-Art

We evaluate our adaptation algorithm against the following baselines that adapt to off-the-shelf pretrained-models without altering the training conditions. *Source-Only* denotes generating the predictions directly with the model trained on the training data, without adaptation. NORM [34] consists in adapting statistics of batch normalization layers to test data. DUA [28] adapts the batch normalization layers online. TENT [42] learns affine transformations of feature maps by minimizing the entropy of test predictions. SHOT (online) [21] maximizes the entropy of batch-wise distribution of predicted classes, while minimizing the entropy of individual predictions. T3A [16] builds pseudo-prototypes from test data.

Batch size 1						
Model	TANet			Swin		
Dataset	UCF101	SSv2	K400	UCF101	SSv2	K400
clean	96.67	59.98	71.64	97.30	66.36	75.32
source	51.35	24.31	37.16	78.48	42.18	47.17
NORM	51.59	17.21	31.40	-	-	-
DUA	<u>55.34</u>	16.20	31.88	-	-	-
TENT	51.58	17.29	31.43	<u>81.19</u>	36.05	45.18
SHOT	51.20	14.54	25.17	68.51	21.32	29.22
T3A	54.17	<u>24.42</u>	<u>37.59</u>	80.66	<u>42.41</u>	<u>48.20</u>
ViTTA	**78.20**	**37.97**	**48.69**	**84.63**	**48.52**	**52.11**

Batch size 8						
Model	TANet			Swin		
Dataset	UCF101	SSv2	K400	UCF101	SSv2	K400
clean	96.67	59.98	71.64	97.30	66.36	75.32
source	51.35	24.31	37.16	78.48	42.18	47.17
NORM	65.77	28.18	39.32	-	-	-
TENT	<u>72.92</u>	<u>31.57</u>	<u>41.13</u>	<u>82.35</u>	<u>42.84</u>	47.84
SHOT	65.54	27.47	37.43	78.42	42.55	47.98
T3A	54.17	24.45	37.59	80.68	42.41	<u>48.20</u>
ViTTA	**78.33**	**38.07**	**48.94**	**84.74**	**49.66**	**54.55**

Table 1. Mean Top-1 Classification Accuracy (%) for batch size 1 and 8 over all corruption types on UCF101, SSv2 and K400. We use the convolution-based TANet and a transformer-based Video Swin Transformer to evaluate our ViTTA and all other baselines. For a fair comparison with baselines, we provide results with different batch sizes, as some baselines require larger batch sizes for optimal performance. *DUA* is only evaluated with batch size of 1, following the setting in the original work [28]. *Clean* refers to the performance of the model on the original validation set of the respective datasets. *Source* is the average performance of the source model over all corruption types without adaptation. Highest accuracy is shown in bold, while second best is underlined.

4.3.1 Evaluation with single distribution shift.

To evaluate the adaptation efficacy of our method and the baselines, we follow [33, 51], and apply each of the 12 corruption types to the entire validation set of all three datasets. We then adapt the networks to each of the 12 resulting corrupted validation sets. We report adaptation results, averaged over the corruption types, in Table 1. Comparing *clean* and *Source-Only*, we see the corruptions drastically deteriorate the performance of both CNN-based TANet and Video Swin Transformer. On adapting the network, the baselines struggle in the online scenario and none of them attains large improvements over the un-adapted model for all three datasets. By contrast, ViTTA yields consistent and significant performance gains in the challenging and practical scenario where videos are received singly.

For a thorough comparison, we also evaluate the adaptation methods with a larger batch size of 8 in Table 1, which is an easier setting for the baseline methods. The baseline methods in general have improvements in comparison to the case of batch size of 1. Note that results of ViTTA with batch size of 1 in Table 1 already surpass the results of all baselines with batch size of 8 to a large margin. As ViTTA accumulates the test statistics in an online manner instead of relying on statistics in a data batch, it does not require a large batch size for good adaptation performance. We further report the results of all the 12 corruption types for adaptation of TANet on UCF101 in Table 2. The results indicate that for most of the corruption types, we outperform the baseline methods to a large margin. More results such as computational efficiency, adaptation on time-correlated data, and adaptation with train statistics from a different dataset can be found in the supplementary.

4.3.2 Evaluation with random distribution shift

We also evaluate the methods for online adaptation in a practical scenario where we assume that each video received has a random type of distribution shift. This scenario might happen when clients in different locations upload videos to the same platform. Specifically, for each of the videos in a sequence, we randomly select one of the 13 distribution shift cases (12 corruption types plus the case of *no corruption*).

This setting is extremely challenging for our online method, since the corruption type changes in each iteration, and the distribution shift between training and test data becomes more complex. This is reflected in the results in Table 3. For many combinations of the dataset and the backbone architecture, the baselines decrease the performance of the un-adapted model. Our method consistently boosts the performance across datasets and architectures. We attribute this robustness against changing corruption types to our technique of aggregating the statistics over multiple adaptation iterations. This decreases the variation in the gradients computed from batches of data with different corruptions.

4.4. Ablation studies

4.4.1 The choice of feature maps to align

Aligning the distribution of the ultimate feature map of a deep network is a common practice in TTA, and in the broader field of domain adaptation [11, 19, 36, 53]. On the other hand, recent work hints that adapting multiple layers in the network [28, 34, 52] might be a more powerful technique, at least when the adaptation is limited to statistics of the batch normalization layers. To verify which of these approaches fares better in our task, we run a series of experiments in which we adapt the TANet and Video Swin transformer to the corrupted UCF101 validation sets. We apply the alignment to the feature map outputs from different combinations of blocks. Here *block* refers to either convolutional *bottleneck* in TANet or *stage* in Swin transformer. We repeat these experiments for four adaptation variants: either by the last block of the architecture, or by

corruptions	gauss	pepper	salt	shot	zoom	impulse	defocus	motion	jpeg	contrast	rain	h265.abr	avg
Source-Only	17.92	23.66	7.85	72.48	76.04	17.16	37.51	54.51	83.40	62.68	81.44	81.58	51.35
NORM	45.23	42.43	27.91	86.25	84.43	46.31	54.32	64.19	89.19	75.26	90.43	83.27	65.77
DUA	36.61	33.97	22.39	80.25	77.13	36.72	44.89	55.67	85.12	30.58	82.66	78.14	55.34
TENT	58.34	53.34	35.77	89.61	**87.68**	59.08	64.92	75.59	90.99	82.53	92.12	**85.09**	72.92
SHOT	46.10	43.33	29.50	85.51	82.95	47.53	53.77	63.37	88.69	73.30	89.82	82.66	65.54
T3A	19.35	26.57	8.83	77.19	79.38	18.64	40.68	58.61	86.12	67.22	84.0	83.45	54.17
ViTTA	**71.37**	**64.55**	**45.84**	**91.44**	**87.68**	**71.90**	**70.76**	**80.32**	**91.70**	**86.78**	**93.07**	84.56	**78.33**

Table 2. Top-1 Classification Accuracy (%) for all corruptions in the UCF101 dataset, while using the TANet backbone. *DUA* is evaluated with batch size of 1, following the setting in the original work [28]. All the other methods are evaluated with batch size of 8.

Model	TANet			Swin		
Dataset	UCF101	SSv2	K400	UCF101	SSv2	K400
Source-Only	55.41	26.93	39.98	79.62	44.31	49.48
NORM	33.32	10.63	27.22	-	-	-
DUA	41.94	12.53	30.89	-	-	-
TENT	31.32	10.68	27.25	81.35	44.58	49.46
SHOT	32.91	9.02	22.89	78.66	32.93	42.37
T3A	53.32	24.74	38.86	81.02	43.54	49.59
ViTTA	**66.94**	**32.87**	**42.76**	**83.11**	**46.32**	**49.67**

Table 3. Mean Top-1 Classification Accuracy (%) with random distribution shifts. For each video in the test set, we randomly select 1 out of 13 distribution shifts (12 corruption types + *original test set*). We run these experiments 3 times while shuffling the order of the videos and report the average results.

Blocks	4	3, 4	2, 3, 4	1, 2, 3, 4
TANet	77.83	**78.20**	76.67	75.37
Swin	82.18	**84.63**	84.47	84.40

Table 4. Mean Top-1 Classification Accuracy (%) over corruptions. We adapt TANet and Video swin tansformer on UCF101 by aligning feature maps from different blocks. *block* refers to either convolutional *bottleneck* in TANet or *stage* in Swin transformer.

the last two, three, or all four blocks. The results, presented in Table 4, suggest that an intermediate approach: aligning feature maps produced by the last two blocks, performs best. We attribute this to the fact that leaving too many degrees of freedom during adaptation might lead to matching the distribution of the last layers, but without transferring feature semantics. On the other hand, some degree of freedom might be needed in the lower layers of the architecture for the network to learn to map appearance of the corrupted data to the feature space of the layers further in the computation graph, learned on the training data without corruption.

4.4.2 Statistics stored in normalization layers

For feature distribution alignment, our method requires feature means and variances computed on the training data. When training data is no longer available, these statistics could be computed on other data with the similar distribution. For architectures that contain batch normalization layers, using the running means and variances, accumulated at these layers during training, represents a convenient alternative. However, these statistics might misrepresent true means and variances of the data, as reported by Wu et al. [47]. To investigate how their inaccuracy affects performance, we compare the results attained by using them as adaptation targets to those obtained by using statistics computed on the training set. We present the results in Tab 5. They confirm that relying on the running mean and variance yields performance slightly lower than computing the statistics from scratch, but still higher than that of the baselines. This demonstrates that our ViTTA can also only rely on batch norm statistics in architectures with BN layers, at a modest performance penalty.

	Statistics	UCF101	SSv2	K400
Source-Only	-	51.35	24.31	37.16
ViTTA	BNS	73.20	35.51	45.30
ViTTA	Src-Computed	**78.20**	**37.97**	**48.69**

Table 5. Mean Top-1 Classification Accuracy (%) over all corruptions on the UCF101 dataset, while using different kinds of source statistics for alignment. We use the TANet backbone for these experiments. BNS refers to the statistics stored in the batch normalization layers. $Src-Computed$ refers to the statistics which we calculate from the source data for distribution alignment.

Views	1	2	2	3	4	5
Pred. Cons.	✗	✗	✓	✓	✓	✓
Acc	75.57	77.46	78.20	78.24	78.25	78.25

Table 6. Mean Top-1 Classification Accuracy (%) over all corruption types for UCF101 by using the TANet backbone. We perform ablation study on different number of temporal views and the prediction consistency regularization.

4.4.3 Number of views and prediction consistency

In temporal augmentation, we compute test statistics on temporally augmented views of the input and enforce pre-

sampling strategy	uniform random	dense random	uniform equidistant	dense equidistant	total random
Accuracy	77.68	76.93	**78.20**	77.10	77.50

Table 7. Mean Top-1 Classification Accuracy (%) over all corruptions on UCF101 using TANet. We perform ablation study on the frame sampling strategies for temporally augmented views.

diction consistency among the views. We verify how much performance gain these design choices yield in Table 6. Without prediction consistency, sampling two temporally augmented views (77.46%) brings around two percent improvement in comparison to only one view (75.57%). Applying the prediction consistency regularization among the two views further adds 0.74% improvement. Sampling more than two augmented views adds little benefit. We set the number of views to two for better adaptation efficiency.

4.4.4 Temporal sampling strategies

Our method relies on temporal augmentation to generate multiple views of the input video sequence and enforce consistency of their predictions. Common temporal sampling methods can be found in video recognition literature [5, 8, 39, 44]. We embed the most representative of these techniques in our adaptation algorithm and evaluate the resulting performance. *Uniform* and *dense* are common strategies for video segment selection. *Random* and *equidistant* are strategies of sampling frames in each video segment. *total random* refers to completely randomly sampling frames from the entire video.

The results in Table 7 show that our ViTTA generalizes well to different types of frame sampling strategies, as they all demonstrate clear performance boost in comparison to the case of one view (75.57%). The *uniform-equidistant* approach perform best. We hypothesize that this stems from the fact that this sampling technique keeps the interval between the video frames constant, while yielding frame sequences that span the entire video sequence, which leads to more accurate statistics of the overall video content.

4.4.5 Continuous adaptation

We check the capacity of the methods to re-adapt to the uncorrupted data. We perform an experiment in the continuous adaptation scenario. When sequentially feeding the test data, we periodically switch the corruption 'on' and 'off' around every five hundred test videos. We use the Gaussian noise as the corruption technique.

As seen in Fig. 3, ViTTA performs constantly the best on corrupted periods. Both our approach and *DUA* can recover the original performance on the un-corrupted periods, while *TENT* performs slightly worse when adapting to the

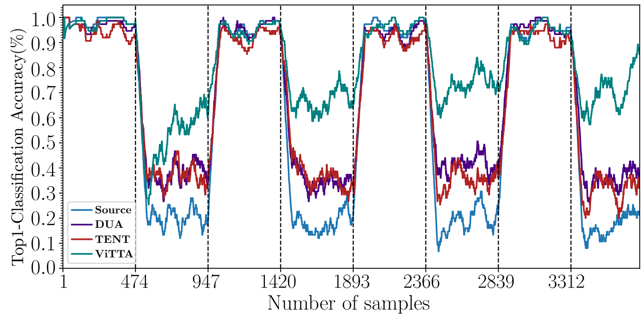

Figure 3. Top-1 Classification Accuracy (%) for a sliding window of 75 samples in a continuous adaptation scenario, where we switch alternatively between Gaussian Noise corruption and the Clean test set of UCF101, we compare with the two best performing baselines *DUA* and *TENT* to avoid clutter. *Source* refers to performance of the pre-trained model without adaptation.

un-corrupted data. *DUA* only corrects the batch norm statistics without updating model parameters. It keeps the knowledge on the un-corrupted but has limited gain in corrupted periods. *TENT* updates the model with an entropy minimization loss and has even slightly worse performance than *Source-Only* in un-corrupted periods. In comparison, our method updates the entire model and still quickly restores the performance in un-corrupted periods, demonstrating the fast reactivity. As our method accumulates the target statistics in an online manner, it also has gradually improved performance when going through the four corrupted periods.

5. Conclusion

We address the problem of test-time adaptation of video action recognition models against common corruptions, We propose a video-tailored method that aligns the training statistics with the online estimates of target statistics. To further boost the performance, we enforce prediction consistency among temporally augmented views of a video sample. We benchmark existing TTA techniques on three action recognition datasets, with 12 common image- and video-specific corruptions. Our proposed method ViTTA performs favorably in evaluation of both, single corruption and the challenging random corruption scenario. Furthermore, it demonstrates fast reactivity on adaptation performance, faced with periodic change of distribution shift.

Acknowledgements We gratefully acknowledge the financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association. This work was also partially funded by the FWF Austrain Science Fund Lise Meitner grant (M3374).

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, pages 6836–6846, 2021. 2, 3

[2] Fatemeh Azimi, Sebastian Palacio, Federico Raue, Jörn Hees, Luca Bertinetto, and Andreas Dengel. Self-supervised test-time adaptation on video data. In *WACV*, pages 3439–3448, 2022. 1

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 2

[4] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free Online Test-time Adaptation. In *CVPR*, 2022. 3, 5

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 2, 8

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 3

[7] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6824–6835, 2021. 2

[8] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 1, 2, 8

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *CVPR*, pages 6202–6211, 2019. 2

[10] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A Efros. Test-time training with masked autoencoders. In *NeurIPS*, 2022. 3

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016. 6

[12] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*, 2018. 3

[13] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, pages 5842–5850, 2017. 2, 5

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders are Scalable Vision Learners. In *CVPR*, 2022. 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[16] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *NeurIPS*, 34:2427–2440, 2021. 1, 5

[17] Yusuke Iwasawa and Yutaka Matsuo. Test-Time Classifier Adjustment Module for Model-Agnostic Domain Generalization. *NIPS*, 2021. 3

[18] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 5

[19] Takeshi Kojima, Yutaka Matsuo, and Yusuke Iwasawa. Robustifying vision transformer without retraining from scratch by test-time class-conditional feature alignment. *arXiv preprint arXiv:2206.13951*, 2022. 1, 2, 4, 6

[20] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *CVPR*, pages 909–918, 2020. 2

[21] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, pages 6028–6039. PMLR, 2020. 1, 3, 5

[22] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, pages 7083–7093, 2019. 2

[23] Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. TTT++: When Does Self-Supervised Test-Time Training Fail or Thrive? In *NeurIPS*, 2021. 1, 3

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 5

[25] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. Teinet: Towards an efficient architecture for video recognition. In *AAAI*, volume 34, pages 11669–11676, 2020. 2

[26] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, pages 3202–3211, 2022. 2, 3, 4, 5

[27] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. In *ICCV*, pages 13708–13718, 2021. 2, 4, 5

[28] M Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *CVPR*, pages 14765–14775, 2022. 1, 2, 3, 4, 5, 6, 7

[29] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020. 4

[30] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *ICCV*, pages 3163–3172, 2021. 2

[31] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *NeurIPS*, volume 34, pages 12493–12506, 2021. 2

[32] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5533–5541, 2017. 2

[33] Madeline C. Schiappa, Naman Biyani, Shruti Vyas, Hamid Palangi, Vibhav Vineet, and Yogesh Rawat. Large-scale robustness analysis of video action recognition models. *CoRR*, abs/2207.01398, 2022. 1, 3, 5, 6

[34] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, pages 11539–11551, 2020. 1, 2, 3, 4, 5, 6

[35] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 5

[36] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, pages 153–171. Springer, 2017. 6

[37] Baochen Sun and Kate Saenko. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. 2016. 2

[38] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. 2020. 1, 3, 5

[39] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *CVPR*, pages 4489–4497, 2015. 2, 8

[40] Thanh-Dat Truong, Quoc-Huy Bui, Chi Nhan Duong, Han-Seok Seo, Son Lam Phung, Xin Li, and Khoa Luu. Direcformer: A directed attention in transformer approach to robust action recognition. In *CVPR*, pages 20030–20040, 2022. 1

[41] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008. 2

[42] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 1, 3, 4, 5

[43] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *CVPR*, pages 1895–1904, 2021. 2

[44] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016. 2, 8

[45] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, pages 7201–7211, 2022. 5

[46] Zhengwei Wang, Qi She, and Aljosa Smolic. Action-net: Multipath excitation for action recognition. In *CVPR*, 2021. 1

[47] Yuxin Wu and Justin Johnson. Rethinking" batch" in batchnorm. *arXiv preprint arXiv:2105.07576*, 2021. 7

[48] Wangmeng Xiang, Chao Li, Biao Wang, Xihan Wei, Xian-Sheng Hua, and Lei Zhang. Spatiotemporal self-attention modeling with temporal patch shift for action recognition. In *ECCV*, 2022. 1

[49] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *CVPR*, 2020. 1

[50] Jiewen Yang, Xingbo Dong, Liujun Liu, Chao Zhang, Jiajun Shen, and Dahai Yu. Recurring the transformer for video action recognition. In *CVPR*, pages 14063–14073, 2022. 1

[51] Chenyu Yi, Siyuan Yang, Haoliang Li, Yap-Peng Tan, and Alex C. Kot. Benchmarking the robustness of spatial-temporal models against corruptions. In *NeurIPS*, 2021. 1, 3, 5, 6

[52] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deep-inversion. In *CVPR*, pages 8715–8724, 2020. 6

[53] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017. 2, 6

[54] Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. Token shift transformer for video classification. In *ACM Multimedia*, pages 917–925, 2021. 2

[55] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *arXiv preprint arXiv:2110.09506*, 2021. 3