ST2ST: Self-Supervised Test-time Adaptation for Video Action Recognition

Masud An-Nur Islam Fahim† , Mohammed Innat⊤, Jani Boutellier†
† University of Vaasa, Finland, ⊤ Khulna University of Engineering & Technology (KUET), Bangladesh
innat.dev@gmail.com, { masud.fahim, jani.boutellier}@uwasa.fi

Abstract

The performance of trained deep neural network (DNN) models relies on the assumption that the test data has largely the same feature distribution as the training data. In deployed video recognition systems, the feature distribution of acquired samples can however become shifted due to environmental conditions (rain, lighting variations) or technological factors such as lossy data compression. To improve action recognition performance under feature distribution shifts, we propose a simple test-time self-distillation strategy where the DNN model goes through an intra-video logit minimization phase. As a result, the model can update its predictions for the given input. The proposed approach is agnostic to the neural network type (CNN, transformer) and applies to various action recognition models. In contrast to many test-time adaption studies, the proposed approach does not require access to the training data. The performance of the proposed method is evaluated with multiple state-of-the-art action recognition models and widely used benchmark datasets Kinetics-400 and Something-Something V2.

1. Introduction

In general, deep neural networks assume clean and i.i.d. training data and achieve remarkable generalization performance if test-time data follows this assumption. However, data from real-world sources often violates these common premises of the training phase. This observation holds for almost all data modalities and is especially prominent in the video domain [10, 20, 39].

For example, video streams used for action recognition are frequently temporally correlated and corrupted due to natural or technical phenomena. Weather effects such as rain, haze, bright daylight, or smoke can affect test-time video capture, together with video technology-related factors (compression, noise, blur, low resolution) can change the statistical properties of videos. Consequently, such distortions cause feature distribution shifts and hamper the performance of video recognition models. These issues are severe, as video recognition models are commonly used in critical systems like autonomous driving and surveillance [10].

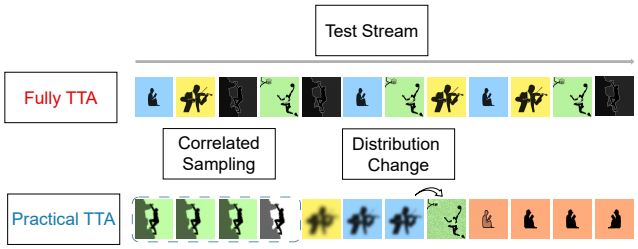

Figure 1. Different test-time adaptation scenarios following the classification of [39]. Fully TTA: test sequence is independently sampled and categories have unchanging feature distribution; Practical TTA: test stream sampling is correlated and undergoes distribution changes (indicated by background color change).

To address issues related to temporal correlation and distribution shifts, adapting models during test time [2, 10, 14, 19, 34, 39, 40] has shown promising results, particularly in the image domain. However, works of the image domain cannot be directly adapted to the video domain for various reasons. For instance, works that leverage entropy minimization [19, 34] focus only on the normalization layers, an approach that does not generalize to transformer-based models [18, 33] that represent the state-of-the-art in action recognition. Additionally, the performance of entropy minimization approaches diminishes in the presence of temporally correlated features, further worsened by continual distribution shifts [39].

In contrast, NOTE [10] and RoTTA [39] approaches show superior performance in the presence of temporal correlation and continual feature shifts; however, they have been designed for *image* data instead of video inputs. TeCo [38] and ViTTA [20] were the first works that studied test-time adaptation (TTA) for video models, with promising results. However, ViTTA [20] requires the distribution of the training data (source domain) to complete its adaptation phase, whereas TeCo [38] works in an offline setup. In a practical TTA setup (see Fig. 1), access to the source domain is not always possible, and online adaptation can be

considered mandatory for a deployed TTA system.

Whereas previous works on video TTA are built on leveraging augmented views of the given inputs [20, 38], our work recognizes that baseline video recognition models have already been pre-trained using extensive data augmentation. Thus application of further regular image augmentation strategies in the TTA phase has a limited impact. In contrast, temporal augmentation strategies such as different sampling intervals, masking, or temporal order shuffling are more effective for self-supervised training video modes [20].

This work presents a novel TTA approach, under the assumption that the incoming video stream is temporally correlated and undergoes continual changes within the feature distribution. We assume the training data is unavailable and the adaptation occurs at test time. The proposed TTA approach leverages *super-frame clips* as a novel spatio-temporal representation of a given video sequence. A super-frame is assembled by spatially concatenating several uniformly sampled frames into a single super-frame [7, 25], and consequently by stacking multiple super-frames, into a *super-frame clip*. Semantically, the input video and the super-frame clip represent the same information, but their logit distributions differ significantly. This discrepancy enables us to define a novel self-supervised training objective, where we aim to minimize the inference gap between the initial corrupted video and the super-frame clip and acquire more accurate predictions for the test streams.

Naive minimization between the input video and the super-frame clip predictions can lead to catastrophic overfitting when the target logits from the corrupted videos have high entropy or when wrong class candidates dominate the extracted logits. The proposed approach mitigates these challenges by batch-based prototype estimation and extends the initial minimization task by setting the prototypes as an additional set of TTA targets. Consequently, our adapted model delivers more accurate predictions for corrupted test-time videos. It needs to be emphasized that during the TTA phase, our source-independent method only requires access to the given video recognition model and to the incoming test stream.

The proposed approach is agnostic towards neural architecture types (transformer, CNN), unlike several previous TTA works. We have validated the proposed approach on multiple state-of-the-art models and datasets, and in all cases, our method significantly improves the baseline action recognition performance. The key contributions of this work are:

- We propose ST2ST, a novel test time adaptation approach for video recognition models, which improves model accuracy under temporally correlated inputs and continually changing distributions.
- Our adaptation algorithm is source-independent, parameter efficient, memory efficient, relies upon batch processing and is model-type agnostic.
- Experimental results are shown for several state-of-the-art action recognition models, and Kinetics-400 and Something-Something V2 datasets.

2. Related Work

Input data corruption of spatio-temporal streams can originate from various sources, including weather phenomena and technological causes such as video compression. Independent of the origin, corruptions lead to changes in the input's feature distribution, reducing the performance of image and video recognition models. The works [28, 37] have benchmarked the performance of video recognition models in the presence of corrupted test data. In our proposed work we investigate the effect of eight types of common corruptions [28, 37] typically observed in the context of video stream acquisition and consequent processing tasks.

Action recognition. Typically, action recognition models are trained on large-scale video datasets such as Kinetics-400 [17], Sports1M [16], or Something-Something V1 or V2 [11]. Related to these large scale datasets, UniFormer-V2 [18] and TubeViT [26] represent state-of-the-art performance, whereas VideoMAE [33] contributes to data-efficient training, and VideoSwin [22, 23] as well as TubeViT [26] on efficient modeling. The proposed work focuses on improving the robustness of video recognition in the case where the incoming temporally correlated video stream undergoes continual changes in the feature distribution [39].

Test-Time Adaptation tunes the model against changes in the input stream's feature distribution and attempts to improve the inference accuracy by un- or self-supervised learning. Recent TTA studies [2, 14, 21, 24, 29, 31, 34] have shown promising performance in the image domain. Among these, one branch of TTA performs classification tasks by means of auxiliary self-supervised minimization [9, 31]. For example, TTT [31] and TTT++ [21] update the baseline classifier through a joint training strategy, where the main task is classification and the auxiliary task concerns self-supervised representation alignment. Similarly, [9] performs rotation prediction for the TTA task. Conversely, the work, [8] updates a masked autoencoder [13] during test time, but is unfortunately not generalizable to other types of networks.

Another set of test-time adaptation studies [14, 19, 34, 40] focuses explicitly on the batch normalization statistics at test time, either for minimizing entropy or for adding supporting regularization on top of entropy loss and updating the model by tuning the batch normalization parameters

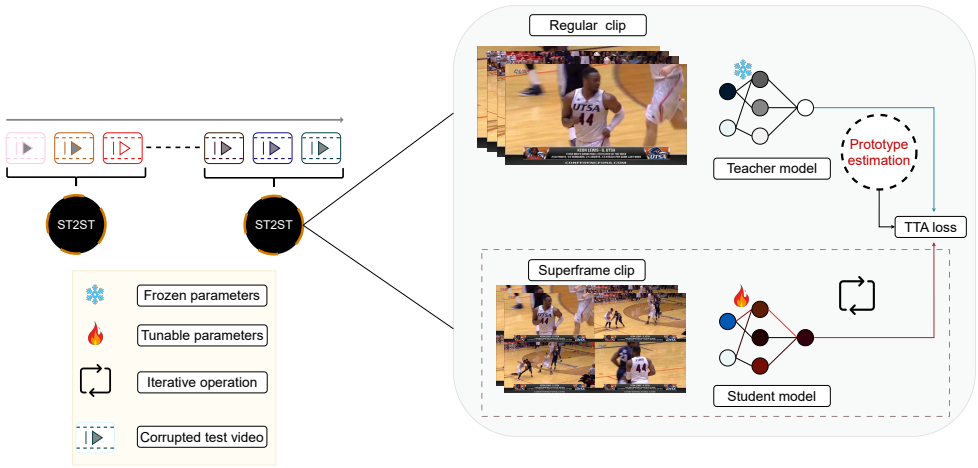

Figure 2. Flowchart of the proposed test-time adaptation scheme.

only. TENT [34] adapts the model by minimizing the entropy of the model predictions. SHOT [19] extends TENT [34] and adds information maximization regularization on top of entropy minimization. MEMO [40] is another extension of TENT [34] that improves on entropy minimization by considering the supporting logits from augmentation strategies.

Some of the TTA works also consider parameter efficiency; for example, LAME [2] does not update the classifier at all. Instead, it offers refined logits through Laplacian optimization based on the baseline predictions. Similarly, T3A [14] updates only the linear layer and returns improved inference by considering prototype logits.

However, all of these methods heavily suffer in performance when the test stream data is temporally correlated. For example, NOTE [10], which is related to our method, deals with non-i.i.d. data streams through instance-aware batch normalization and prediction-balanced reservoir sampling. Unfortunately, NOTE [10] does not manage well continually changing non-i.i.d. streams [39]. In contrast, RoTTA [39] addresses test-time adaptation scenarios where the feature distribution is temporally correlated and changes continually; this is achieved by a robust batch normalization layer and a logit memory bank. Our proposed approach considers a similar input data scenario as [39], which is common for autonomous driving, video-assisted broadcasting, or surveillance applications. However, whereas RoTTA [39] concentrates on image recognition, the proposed approach targets video recognition.

TTA has gained attention in the video recognition domain only recently, through two works: TeCo [38] provides TTA through different gradient update policies by entropy minimization and temporal coherence regularization. Here, authors [35] tried TTA one segmentation task with videos using masked image modeling. ViTTA [20] requires input video statistics to update normalization parameters during test time on top of temporal augmentation-guided regularization. In contrast to these works, the proposed approach is not layer-specific nor requires source distribution information during the TTA task.

Notation	Description
$\boldsymbol{x}$	Test video
$\boldsymbol{y}$	Unknown clean label
$\boldsymbol{\mathcal{D}}$	Test stream
$\hat{\boldsymbol{y}}$	noisy label
$\mathcal{B}$	Sampled corrupted batch
$\mathcal{F}_\theta$	Pre-trained action classifier
$\boldsymbol{\mathcal{F}_{\theta^*_{TTA}}}$	Test-time adapted model
$\boldsymbol{\mathcal{H}}$	Information estimator
$\boldsymbol{c_i}$	Sampled clip from a video
$\mathcal{T}$	clip length
$\boldsymbol{c_i^r}$	Superframe
$\boldsymbol{x_n^r}$	Superframe clip
$\boldsymbol{\mathcal{B}_{y^P}}$	prototype set for $\mathcal{B}$
Sum	Summation
Del	Delete specific entry
$Copy$	Copy specific entry

Table 1. Notations index for this study.

3. The Proposed Video TTA Approach

Here, we describe our TTA approach for action classification, especially when the incoming stream is temporarily correlated. Let $\mathcal{D}$ represent a temporally correlated and continually changing test-time video stream such that $\mathcal{D} = [(x_1, y_1), ..., (x_n, y_n)] = (X, Y)$, where (x_n, y_n) denotes a random video-label pair, and label set Y is unknown. A pre-trained action classifier model $\mathcal{F}_\theta$ is used for inferring the logit set $\hat{Y} = \mathcal{F}_\theta(X)$ at test time. We assume

that the incoming stream $\mathcal{D}$ is corrupted by unknown distribution shifts, and hence the predicted logits $\hat{Y}$ are noisy. After initial inference by $\mathcal{F}_\theta$ it is possible to create a noisy paired dataset, $\hat{\mathcal{D}} = [(x_1, \hat{y_1}), ..., (x_n, \hat{y_n})] = (X, \hat{Y})$ for test time adaptation, and to obtain the test-time adapted model $\mathcal{F}_{\theta^*_{TTA}}$, which is expected to provide more precise predictions. A generic TTA approach aims to minimize the following objective:

$$\mathcal{F}_{\theta^*_{TTA}} = \underset{\mathcal{F}_\theta}{\operatorname{argmin}} \sum \mathcal{L}_{TTA}\left(\mathcal{F}_\theta; \hat{\mathcal{D}}\right) \tag{1}$$

that is applicable both at training and test time. $\mathcal{L}_{TTA}$ in Eqn. 1 can represent, e.g., a self-supervised distillation [36], entropy minimization [34, 38], or contrastive learning approach [4, 27, 32]. Naive distillation by $\mathcal{F}_\theta$ would return an identity solution, whereas regular entropy minimization [34] would not provide an implementation faithful to the original implementation since $\mathcal{F}_\theta$ can be independent of batch normalization (e.g., transformers/variants [18,33]), and the original [34] implementation tunes batchnorm parameters during TTA. Since our study addresses the scenario of temporally correlated feature distributions, adopting entropy minimization would lead to model degeneration [2] or catastrophic forgetting [39].

Contrastive or self-supervised approaches could provide a generic solution to the targeted test-time adaption task, but inference complexity can increase prohibitively depending on the model or augmentation strategy (e.g., [4] requires multiple forward and backward passes to extract necessary logits and following TTA phase; with videos, this strategy necessitates *C* folds computational increments as video models require *C* clips per video). For video recognition tasks, low computational complexity is desirable due to the task's stream-processing nature and clip dependencies. Finally, common data augmentation strategies that work for contrastive tasks in image recognition, provide limited impact as many video recognition models are inflated versions of image classifiers [3] that have already been pre-trained with usual data augmentation approaches.

Motivated by the above, we propose *Spatiotemporal to Spatiotemporal* (ST2ST), a self-supervised test-time adaption method for video action classification. This self-supervised minimization task utilizes a lightweight contrastive operation that provides a robust learning incentive to video recognition models. A detailed description of the proposed method is provided in the following subsection, whereas Table 1 explains our notation.

3.1. Spatiotemporal to Spatiotemporal

At the inference stage of a conventional action classifier, a test-time video x_n is sampled and segmented into N small clips, passed to the action classifier model $\mathcal{F}_\theta$, followed by averaging of the clip-wise predictions and matched against the ground truth y_n.

Algorithm 1: TTA with Spatiotemporal to Spatiotemporal

Input: Test batch logits $\mathcal{B}_{\hat{y}}$, Superframe clip x^r_n, pretrained model $\mathcal{F}_\theta$, adaptation epoch ep, Convex weights $w_1 \& w_2$

Output: Adapted model $\mathcal{F}_{\theta^*_{TTA}}$ for the given batch

Initialize teacher model $\mathcal{F}_{\theta t}$, and freeze it ;
$\mathcal{B}_{\hat{y}} \leftarrow \mathcal{F}_{\theta t}(x_n)$;
$\mathcal{B}_{y^P} \leftarrow$ Algorithm 2;
Initialize adapted model $\mathcal{F}_{\theta^*_{TTA}}$;
Set $w_1 = 0.9$ & $w_2 = 1 - w_1$;
for 1 **to** ep **do**
  $\mathcal{F}_{\theta^*_{TTA}} \leftarrow \mathcal{L}(\mathcal{F}_{\theta^*_{TTA}}, x^r_n, \mathcal{B}_{y^P}, \mathcal{B}_{\hat{y}})$, Eqn. 2
end
Return adapted model $\mathcal{F}_{\theta^*_{TTA}}$

For a given video x_n, we uniformly sample N clips such that $x_n = [c_1, c_2, .., c_i, .., c_N]$, and each clip is of the same length denoted by $\mathcal{T}$. The amount of spatiotemporal information $\mathcal{H}(x_n)$ in the clips can be expressed as $\mathcal{H}([c_1, c_2, ..., c_N])$. Next, each clip c_i is converted into a *superframe* image c^r_i that approximates whole clip c_i, and the spatiotemporal information within c_i becomes embedded into a single spatial-domain image c^r_i. In terms of information content, $\mathcal{H}(c_i) \cong \mathcal{H}(c^r_i)$. We restrict ourselves to maintain the superframe c^r_i spatial resolution identical to the original frame c_i spatial resolution, and hence we encode only 4 frames into a superframe c^r_i (see Fig. 1) instead of $\mathcal{T}$, whereas $\mathcal{T} = [8, 16, 32]$ as required by the underlying video recognition model [18, 23, 33].

By repeating the same operation for all the clips $[c_1, .., c_N]$, the whole video can be represented as a sequence of superframes: $x^r_n = [c^r_1, c^r_2, .., c^r_i, .., c^r_N]$. Here, x^r_n is a novel spatiotemporal representation of the original video x_n. The original N clips have now been encoded into a single clip x^r_n with roughly the same information content $\mathcal{H}(x_n) \cong \mathcal{H}(x^r_n)$. Whereas in the original video representation x_n each clip consists of temporally sampled frames, in the superframe representation x^r_n the interleaving of frames has been reshaped into a spatial interleaving such that there is one superframe $c^r_i \in x^r_n$ for each original clip c_i. Performing action recognition for x^r_n using our pre-trained model $\mathcal{F}_\theta$ (e.g. VideoSwin [23]) yields significantly reduced performance (see Table 11) compared to inference on the conventional input $\mathcal{F}_\theta(x_n)$. Yet, intuitively it is clear that both x^r_n and x_n bear roughly the same information content. This leads us to the central question: *can the prediction discrepancy between $\mathcal{F}_\theta(x^r_n)$ and $\mathcal{F}_\theta(x_n)$ be leveraged to train a more generalizable model?*

We consider $\mathcal{B}$ as a given batch of test-time videos and

prepare $\mathcal{B}^r$ as the corresponding superframe representation of $\mathcal{B}$. From the pre-trained model $\mathcal{F}_\theta$ we initialize a teacher model $\mathcal{F}_{\theta t}$ by copying the weights of $\mathcal{F}_\theta$, and freeze the teacher. From the teacher model, we extract $\mathcal{B}_{\hat{y}} = \mathcal{F}_{\theta t}(x_n)$ as the initial logits or pseudo labels. Similarly, we initialize the TTA / student model $\mathcal{F}_{\theta^*_{TTA}}$ from $\mathcal{F}_\theta$, but let the student model be trainable. Thus, the proposed self-supervised minimization objective is $\boldsymbol{\mathcal{L}_1} = \left\|\mathcal{F}_{\theta^*_{TTA}}(\mathcal{B}^r) - \mathcal{B}_{\hat{y}}\right\|_2^2$.

Here, $\mathcal{B} = [x_1, x_2, .., x_m]$, $\mathcal{B}^r = [x_1^r, x_2^r, .., x_m^r]$, and m is the batch size. As our objective function operates on soft labels, we consider mean-square error (MSE) minimization as the target. By updating the test-time adaptive model $\mathcal{F}_{\theta^*_{TTA}}$ over a few epochs, we acquire the final $\mathcal{F}_{\theta^*_{TTA}}$ for the current batch $\mathcal{B}$. On overview of the proposed TTA scheme is shown in Fig. 2.

Even though the test time adaptation described above already improves model accuracy, robustness against noisy $\mathcal{F}_\theta(x_n)$ pseudo labels can be achieved by prototype supervision, as explained in the following subsection.

3.2. Prototype supervision

Leveraging class prototypes has become a popular approach in noisy label distillation [6, 12, 41] and in test time adaptation [1, 15, 36, 39] although estimating prototypes from noisy streams is not straightforward. TTA methods with prototype estimation generally resort to memory banks [39] or exponential moving average (EMA) [5] operations to estimate prototypes on the fly. Unfortunately, these approaches introduce extra computations that increase as a function of class count.

More importantly, traditional video recognition models expect a single video instance to be represented as a set of clips, and the final inference result is the average of clip-wise predictions. Hence, traditional prototype cost increases N folds, where N is the number of clips and prototypes that need to be refined over time [36, 39].

We assume the same test stream characteristics, non-i.i.d and continually changing distribution, as previous works [10, 39]. Due to correlated sampling, consecutive classes in the input stream have a tendency to originate from the same class (see Fig. 1), for a batch of videos. Consequently, the prototype set can be established from the batch members such that $\mathcal{B}_{\hat{y}} = \mathcal{F}_{\theta t}(x_n)$ is the initial logit set for a given batch. Naive averaging between all the members from $\mathcal{B}_{\hat{y}}$ can provide a class representative for that particular batch. The solution is obviously not perfect, as outlier logits or sudden class changes can cause failing prototype estimation.

In Algorithm 2 for prototype estimation, we follow the convex averaging procedure for extracting the prototype for each batch member. For a single video x_n of batch $\mathcal{B}$, the prototype q_n is obtained via a convex-average between the self logit P_1 and the batch neighbor logits P_2, where higher weight is assigned to P_1. Since the prototypes are batch-based, they do not affect the outcome of future batches.

Algorithm 2: Pseudocode for batch prototype estimation

Input: Test batch logits $\mathcal{B}_{\hat{y}}$, convex weights w_1, w_2
Output: $\boldsymbol{q}$, Prototype set for the given batch $\mathcal{B}$

Initialize prototype set as $\boldsymbol{q}$;
Set $w_1 = 0.9$ & $w_2 = 1 - w_1$;
`/* Length of` $\mathcal{B}$`,` L `*/`
for $l \leftarrow 1$ **to** L **do**
  $\mathcal{B}_1$ = Copy($\mathcal{B}_{\hat{y}}$)
  P_1 = $\mathcal{B}_1[l]$
  P_2 = Sum(Del($\mathcal{B}_1[l]$))
  $P_1 = w_1 * P_1 + w_2 * P_2$
  $\boldsymbol{q} \leftarrow P_1$
end

The proposed label prototype estimation approach is lightweight compared to conventional approaches [1, 15, 36, 39] that use a memory bank or global updating policy, which require observing all samples and resorting to matching procedures due to temporally uncorrelated samples.

Now, we denote prototype set $\boldsymbol{q}$ as $\mathcal{B}_{y^P}$, and include it as an additional supervised objective, $\boldsymbol{\mathcal{L}_2} = \left\|\mathcal{F}_{\theta^*_{TTA}}(\mathcal{B}^r) - \mathcal{B}_{y^P}\right\|_2^2$. The complete formation of our minimization objective is as follows:

$$\mathcal{L}_{TTA} = \mathcal{L}_1 + \mathcal{L}_2 \tag{2}$$

Due to including class prototypes in the objective function, our test time adaptation algorithm delivers increased robustness, as well as reduced overfitting across our experiments. By following Algorithm 1, ST2ST minimizes Eqn. 2 in an iterative manner. Determined empirically, ST2ST needs at least five iterations per batch to deliver improved adaptation performance (Section 5.5 details on this).

Rationale of the proposed approach. Previous TTA works build on normalization statistics correction [10, 34, 39] or generic contrastive design [4, 13, 27]. However, neither of these approaches is agnostic to the action recognition model architecture, moreover, the latter is not always effective for video modalities. By including the *Spatiotemporal to Spatiotemporal* transformation (Section 3.1), it is possible to impose an efficient adaptation phase, where the pre-trained model is fine-tuned at test time over a novel spatiotemporal representation. The adaptation phase is amplified further by prototype supervision that mitigates the effect of outliers [41] and the impact of noisy labels [6, 12].

4. Experiments

We present experimental results to evaluate the proposed test-time adaptation approach in the following. The action

recognition models and datasets used are briefly described below.

Datasets: For evaluation we have used two well-known and recent action recognition datasets, Kinetics-400 [17] and Something-Something V2 [11]. Kinetics-400 comes with roughly 240K training and 20K validation videos. Something-Something V2 provides around 168K training videos and 24K validation videos.

Models: The proposed test-time adaptation was applied on top of several state-of-the-art video recognition models: VideoMAE [33], UniFormer-V2 [18] and VideoSwin [23]. The author-provided weights of each model were used for all the experiments and as a pre-check the published performance scores of each model were successfully reproduced on a distortion-free dataset. For the ablation studies, VideoMAE [33] and VideoSwin [23] were mostly used. It is necessary to remark that the VideoSwin [23] model requires 32 frames per clip in inference. In our experiments, the 32-frame input was used only for benchmarking baseline VideoSwin [23] performance against various corruptions (Table 4). In the experiments that involve TTA, 8 frames per clip have been used due to GPU memory restrictions. Our TTA algorithm follows per-batch adaptation policy across all the experiments.

Corruptions: For representing a corrupted test stream with shifted features, we followed the experimental setting of [20] and applied all the same corruptions on the test streams as [20]. All of these were applied to both Kinetics-400 [17], and Something-Something V2 [11].

Baselines: To demonstrate the DNN model architecture independence of the proposed test-time adaptation approach, the experiments have been conducted on several models that consist of convolution, attention, and attention-convolution blocks. As discussed earlier, several previous TTA algorithms have relied on batch normalization statistics adaptation. In the case of batch norm-free architectures, application of such a TTA approach leads to inconsistencies; for instance, [19, 29, 34] do not follow the original implementation with the VideoMAE [33] model as it does not include batch normalization layers. In contrast, the VideoSwin model enabled comparing the proposed approach against BN [29], DUA [24], TENT [34], SHOT [19], T3A [14], and ViTTA [20]. At the time of writing, available *video* test-time adaptation works consist of ViTTA [20] and TeCO [38]. However, ViTTA [20] reports only SwinFormer performance, and the implementation of TeCO [38] has not yet been published.

4.1. Main results on test-time adaptation

The main experiments on test-time adaptation were implemented by sampling batches and clips from temporally correlated test streams by temporally uniform sampling. To introduce temporal correlation, we followed temporally correlated sampling using Dirichlet distribution parameter δ [10, 39], and for our study, we fixed $\delta = 0.001$. The aforementioned corruptions were applied to all of the test samples to simulate real-life conditions. The number of clips for a single video was determined by the underlying model (e.g., SwinFormer or VideoMAE), and a single spatio-temporal clip was extracted for each video. During the TTA phase, the Adam optimizer was used without a scheduler, and the learning rate was 0.001. After adaptation with the current batch, an *adapted model* was provided, which was used to extract refined predictions for the same batch.

Method	Inference	Frames	Kinetics-400	SSV2
VideoSwin [23]	Baseline	32	47.17	42.18
	ST2ST	8	**54.57**	**53.24**
VideoMAE [33]	Baseline	16	54.11	42.04
	ST2ST	16	**71.59**	**59.10**
UniFormer-V2 [18]	Baseline	16	49.30	44,48
	ST2ST	16	**73.40**	**56.23**

Table 2. Corrupted video classification accuracy (Top-1) for [18, 23, 33] with and without ST2ST test-time adaptation. For VideoSwin input frame count, see Models in Section 4.

The main result of the proposed work is shown in Table 2, where the classification performance of the baseline model and its test-time adapted version are shown for VideoSwin [23], VideoMAE [33], and UniFormer-V2 [18] on the Kinetics-400 [17] and Something-Something V2 [11] datasets. Table 2 shows the consistent performance improvement provided by the proposed approach across all models and datasets.

Table 3 compares the test-time adaptation performance of the proposed approach against previous works on the VideoSwin model. As explained earlier in this section, it was not possible to compare the proposed approach against previous works on other models than VideoSwin due to the model architecture or data modality (image) specificity of other works. The results show that the proposed ST2ST approach provides superior performance compared to previous works. The only exception is provided by ViTTA which is outperformed by only a slight margin. In the case of ViTTA, it is however important to point out that the proposed work leverages only 8-frame clips whereas ViTTA performs classification based on 32 frames.

5. Ablation Studies

This section provides several complementary results that address and evaluate different aspects of the proposed test-time adaptation approach. For ablations, subset of 2900 videos from both Something-Something V2 [11] and Kinetics-400 [17] are used.

Method	#Clips	#Frames	SSV2	Kinetics-400
Baseline	3	32	42.18	47.17
Tent [34]	-	32	42.84	47.84
SHOT [19]	-	32	42.55	47.98
T3A [14]	-	32	42.41	48.20
ViTTA [20]	3	32	49.66	54.55
Ours	3/1	8	**53.24**	**54.57**

Table 3. VideoSwin [23] action classification accuracy with and without test-time adaptation approaches, including the proposed one, ST2ST. Here, experiments were conducted on the whole test set of Kinetics-400 and SSV2. For ST2ST, $3/1$ clip count means inference on the three clips from x_n or with a single superframe clip x_n^r after the TTA step (see Subsection 5.6). Clip counts for Tent, SHOT and T3A are not shown as their papers [14, 19, 34] did not provide this information.

Corruption type	Parameter range	Baseline	Adapted
Zoom	1.0 - 2.0	37.18	**52.11**
Brightness	0.1 - 0.3	46.53	**62.89**
Saturation	1.0 - 8.0	41.88	**56.81**
Contrast	0.3 - 0.9	40.35	**61.22**
Resolution	1 - 4	38.35	**59.35**
Noise	25 - 55	30.72	**41.79**
Blur	3 - 19	37.19	**55.02**
JPEG	40 - 80	35.49	**56.99**

Table 4. ST2ST test-time adaptation performance for single-source corruption on a subset of 2900 videos from Something-Something V2 [11] using the VideoSwin [23] model.

Method	Inference	Kinetics-400	SSV2
VideoSwin [23]	$\mathcal{L}_1$	46.32	44.74
	$\mathcal{L}_1 + \mathcal{L}_2$	**53.17**	**49.92**
VideoMAE [33]	$\mathcal{L}_1$	69.57	61.33
	$\mathcal{L}_1 + \mathcal{L}_2$	**73.14**	**65.76**

Table 5. Impact of *prototype supervision* on the proposed test-time adaptation algorithm. Note that the above result was achieved with only a subset (2900 videos) of both datasets [11, 17].

5.1. Individual types of data corruption

Our first ablation presents the test-time adaptation performance when only a single corruption source is applied at a time to Something-Something V2 [11]. Results are shown in Table 4 for each corruption type. In all cases, our adaptation strategy significantly improves the performance of the VideoSwin model [23].

5.2. Impact of prototype supervision

Besides the spatio-temporal representation transformation (Section 3.1), the proposed approach also uses prototype supervision (Section 3.2), where the prototypes are extracted from the corrupted videos using the given video model.

To evaluate the performance impact of this additional supervision objective, Table 5 shows the classification accuracy with and without prototype supervision for the proposed approach. The results indicate that prototype supervision has a significant performance impact.

Method	# Frames	$ep = 4$	$ep = 6$	$ep = 8$
VideoMAE [33]	16	56.67	66.81	73.14
VideoSwin [23]	8	43.19	45.66	51.11

Table 6. The effect of iteration count on the ST2ST algorithm. Increasing the iteration count gradually improves accuracy at the expense of latency. Likewise, [33] shows the results for Kinetics-400 [17], and [23] is for the Something-Something V2 [11].

Method	no TTA	10%	50%	90%
VideoMAE [33]	52.14	59.43	63.65	71.06
VideoSwin [23]	41.09	42.77	44.16	53.20

Table 7. Percentage of test-time tunable model parameters vs. adaptation performance for ST2ST. Similar to the ST2ST iteration count, increasing the number of tunable parameters increases both performance and latency.

5.3. ST2ST adaptation iterations

As mentioned in Section 3.2, ST2ST works in an iterative manner. Through experimentation, a reciprocal relation between TTA iteration count ep and accuracy was observed, as well as between iteration count and latency. Table 6 shows ST2ST classification accuracy as a function of ep. In all experiments, $ep = 6$ was used.

5.4. Trainable parameters vs. performance

Video recognition models typically have many trainable parameters, which come with a significant inference cost. At test-time adaptation, backpropagating through the whole model further increases the computational load. Even though some of the previous TTA works [14, 19, 34] update only the batch normalization layers, they also need to store the intermediate activations of the model for gradient computation, and are therefore also affected by computation cost and storage impacts [30].

The proposed ST2ST approach has the potential to update each trainable parameter of the given model, which also makes computation and memory cost discussion relevant. However, due to the flexibility of our approach, the cost of ST2ST can be reduced by applying it only to a subset of the trainable model parameters; Table 7 shows results where only a fraction of the model parameters are exposed to test-time adaptation; when ST2ST is applied to even a fraction of all model parameters, accuracy increase is clearly observable.

5.5. Effect of batch size

The proposed test time adaptation algorithm leverages batch processing, and consequently, larger batch sizes enable a higher number of temporal and class-correlated samples and enable ST2ST to achieve more accurate prototypes

Method	#Clips	#Frames	$\mathcal{B}=2$	$\mathcal{B}=3$	$\mathcal{B}=4$
VideoMAE [33]	4	16	55.03	61.97	69.21
VideoSwin [23]	4	8	43.71	47.07	51.33

Table 8. The effect of batch size on the proposed ST2ST algorithm, where the top row is showing the result for Kinetics-400 [17] and the bottom is for Something-Something V2 [11].

Method	Inference	Clips	Kinetics-400	SSV2
VideoSwin [23]	Regular x_n	4/3	54.57	53.24
	Superf. x_n^r	1	**54.57**	**53.24**
VideoMAE [33]	Regular x_n	5/3	76.76	59.10
	Superf. x_n^r	1	**76.76**	**59.10**

Table 9. Forward pass accuracy comparison between the *superframe* clip x_n^r and the regular video x_n. After the TTA, $\mathcal{F}_{\theta^*_{TTA}}$ needs only **one** forward pass to get the final logits from x_n^r, whereas 3 to 5 forward passes are needed if multi-clip averaged logits from x_n are used.

than small batch sizes.

Table 8 shows that an increase in batch size consistently improves adaptation results. Larger batch size also contributes to improved predictions, which can be observed for both VideoMAE and VideoSwin models. Due to GPU memory constraints, batch size was restricted to a maximum of four.

5.6. Using the superframe clip x_n^r for post-TTA inference

The use of TTA methods (at least) doubles the runtime cost of a recognition model, as TTA generally needs to have access to pre- and post-TTA logits. For a video recognition model, this issue is even more severe, as it is customary to segment the input video into *N* clips (depending on the model) and average their logits for acquiring the final prediction. For example, VideoMAE [33] requires at least three forward passes before and after TTA to get the initial and adapted logits, totaling six forward passes per video.

With ST2ST, it is possible to reduce the forward pass cost by switching to *superframe* clip inference mode after the TTA. The proposed algorithm minimizes the logit discrepancy between superframe clip x_n^r and the regular clip set x_n, followed by providing the adapted model $\mathcal{F}_{\theta^*_{TTA}}$. In practice, $\mathcal{F}_{\theta^*_{TTA}}$ returns the same logits for both x_n^r and x_n, but requires generally a higher forward pass cost with x_n (depending upon the action recognition model). Table 9 shows that the proposed ST2ST approach can also offer reduced inference latency without any accuracy impact.

5.7. Effect of Dirichlet concentration parameter

The Dirichlet concentration parameter δ is used in our work (similar to [39]) to represent the degree of correlation among test samples; a smaller value of δ represents greater temporal correlation within the test distribution. Table 10 shows how the performance of ST2ST changes as a function of δ from 0.1 to 0.001 in the case of Kinetics-400 and Something-Something V2 datasets and underlying models VideoMAE and VideoSwin.

Method	Dataset	$\delta=0.1$	$\delta=0.01$	$\delta=0.001$
VideoMAE [33]	Kin.-400 [17]	65.79	69.42	71.65
VideoSwin [23]	SSV2 [11]	45.12	49.76	51.33

Table 10. The effect of the Dirichlet concentration parameter δ. By varying δ, the non-i.i.d. behavior within the test distribution. Results are shown for [33] with a subset of Kinetics-400 [17], and in the case of [23] for a subset of Something-Something V2 [11]

Method	Dataset	x_n^r	x_n
VideoMAE [33]	Kinetics-400 [17]	34.89	**77.86**
VideoSwin [23]	SSV2 [11]	25.74	**65.17**

Table 11. Baseline model (no test-time adaptation) Top-1 classification performance using superframe clips x_n^r and regular clips x_n for [23, 33] models for a subset of 2900 videos from Kinetics-400 and Something-Something V2.

5.8. Generalization with superframe clips

The *superframe* clip x_n^r is one of the cornerstones of ST2ST and provides a spatiotemporal representation that regular action classifiers cannot treat well without adaptation. Table 11 quantifies this claim, showing that an action classifier model trained only on regular clips x_n suffers more than 30% in Top-1 classification accuracy if it is provided with superframe clips x_n^r at test time. In contrast, when ST2ST adaptation is performed at test time only for a few iterations (See Table 6), the models quickly adapt to the new representation and, more importantly, improve in overall generalizability (see Table 2).

6. Conclusion

This paper proposes ST2ST, a test-time adaptation method for temporally correlated and corrupted videos that undergo continual feature shifts. ST2ST aligns the spatiotemporal semantics between the input video and its corresponding *superframe clip* representation at test time and adapts the underlying video recognition model to refine its initial predictions via self-supervised distillation. ST2ST is robust against various types of corruption and agnostic towards deep neural architecture designs. Additionally, it can perform TTA without having access to the raw source data and is parameter-efficient. The performance of ST2ST has been validated using multiple action recognition models and state-of-the-art datasets.

Acknowledgements

This work has been partially funded by the Academy of Finland project SPHERE-DNA.

References

[1] Alexander Bartler, Florian Bender, Felix Wiewel, and Bin Yang. Ttaps: Test-time adaption by aligning prototypes using self-supervision. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022. 5

[2] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free Online Test-time Adaptation. In *CVPR*, 2022. 1, 2, 3, 4

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 4

[4] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022. 4, 5

[5] Sungha Choi, Seunghan Yang, Seokeon Choi, and Sungrack Yun. Improving test-time adaptation via shift-agnostic weight regularization and nearest source prototypes. In *European Conference on Computer Vision*, pages 440–458. Springer, 2022. 5

[6] Yuhe Ding, Lijun Sheng, Jian Liang, Aihua Zheng, and Ran He. Proxymix: Proxy-based mixup training with label refinery for source-free domain adaptation. *Neural Networks*, 167:92–103, 2023. 5

[7] Quanfu Fan, Rameswar Panda, et al. Can an image classifier suffice for action recognition? *arXiv preprint arXiv:2106.14104*, 2021. 2

[8] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A Efros. Test-time training with masked autoencoders. In *NeurIPS*, 2022. 2

[9] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*, 2018. 2

[10] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022. 1, 3, 5, 6

[11] 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, 6, 7, 8

[12] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5138–5147, 2019. 5

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders are Scalable Vision Learners. In *CVPR*, 2022. 2, 5

[14] Yusuke Iwasawa and Yutaka Matsuo. Test-Time Classifier Adjustment Module for Model-Agnostic Domain Generalization. *NIPS*, 2021. 1, 2, 3, 6, 7

[15] Minguk Jang, Sae-Young Chung, and Hye Won Chung. Test-time adaptation via self-training with nearest neighbor information. *arXiv preprint arXiv:2207.10792*, 2022. 5

[16] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 2

[17] 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, 6, 7, 8

[18] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *arXiv preprint arXiv:2211.09552*, 2022. 1, 2, 4, 6

[19] 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, 2, 3, 6, 7

[20] Wei Lin, Muhammad Jehanzeb Mirza, Mateusz Kozinski, Horst Possegger, Hilde Kuehne, and Horst Bischof. Video test-time adaptation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22952–22961, 2023. 1, 2, 3, 6, 7

[21] 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. 2

[22] 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. 2

[23] 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, 4, 6, 7, 8

[24] 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. 2, 6

[25] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Dual-path adaptation from image to video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2203–2213, 2023. 2

[26] AJ Piergiovanni, Weicheng Kuo, and Anelia Angelova. Rethinking video vits: Sparse video tubes for joint image and video learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2214–2224, 2023. 2

[27] Aadarsh Sahoo, Rutav Shah, Rameswar Panda, Kate Saenko, and Abir Das. Contrast and mix: Temporal contrastive video domain adaptation with background mixing. In *NeurIPS*, 2021. 4, 5

[28] 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. 2

[29] 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. 2, 6

[30] Junha Song, Jungsoo Lee, In So Kweon, and Sungha Choi. Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11920–11929, 2023. 7

[31] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. In *Int. Conf. Machine Learn.*, 2020. 2

[32] Devavrat Tomar, Guillaume Vray, Behzad Bozorgtabar, and Jean-Philippe Thiran. Tesla: Test-time self-learning with automatic adversarial augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20341–20350, 2023. 4

[33] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022. 1, 2, 4, 6, 7, 8

[34] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 1, 2, 3, 4, 5, 6, 7

[35] Renhao Wang, Yu Sun, Yossi Gandelsman, Xinlei Chen, Alexei A Efros, and Xiaolong Wang. Test-time training on video streams. *arXiv preprint arXiv:2307.05014*, 2023. 3

[36] Shuai Wang, Daoan Zhang, Zipei Yan, Jianguo Zhang, and Rui Li. Feature alignment and uniformity for test time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20050–20060, 2023. 4, 5

[37] 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. 2

[38] Chenyu Yi, Siyuan Yang, Yufei Wang, Haoliang Li, Yap-Peng Tan, and Alex C Kot. Temporal coherent test-time optimization for robust video classification. *arXiv preprint arXiv:2302.14309*, 2023. 1, 2, 3, 4, 6

[39] Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15922–15932, 2023. 1, 2, 3, 4, 5, 6, 8

[40] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *arXiv preprint arXiv:2110.09506*, 2021. 1, 2, 3

[41] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12414–12424, 2021. 5