Self-supervised Test-time Adaptation on Video Data

Fatemeh Azimi[⋆†] Sebastian Palacio[⋆†] Federico Raue[⋆] Jörn Hees[⋆] Luca Bertinetto[‡] Andreas Dengel[⋆†]
[⋆] German Research Center for Artificial Intelligence (DFKI)
[‡] Five AI Ltd.
[†] TU Kaiserslautern
{fatemeh.azimi,sebastian.palacio,federico.raue,andreas.dengel}@dfki.de, luca@robots.ox.ac.uk

Abstract

In typical computer vision problems revolving around video data, pre-trained models are simply evaluated at test time, without adaptation. This general approach clearly cannot capture the shifts that will likely arise between the distributions from which training and test data have been sampled. Adapting a pre-trained model to a new video encountered at test time could be essential to avoid the potentially catastrophic effects of such shifts. However, given the inherent impossibility of labeling data only available at test-time, traditional "fine-tuning" techniques cannot be leveraged in this highly practical scenario. This paper explores whether the recent progress in test-time adaptation in the image domain and self-supervised learning can be leveraged to adapt a model to previously unseen and unlabelled videos presenting both mild (but arbitrary) and severe covariate shifts. In our experiments, we show that test-time adaptation approaches applied to self-supervised methods are always beneficial, but also that the extent of their effectiveness largely depends on the specific combination of the algorithms used for adaptation and self-supervision, and also on the type of covariate shift taking place.

1. Introduction

Most modern computer vision applications follow the general two-steps paradigm of first training a model on a large dataset and then deploying it on unseen test data. However, the majority of these applications are still designed under the assumption that training and test data have been sampled from the same distribution. As this assumption is frequently violated in the real-world, the applicability of these models can often be very limited [11, 30] It is thus important to seek strategies for adapting pre-trained models to the test data. However, *supervised* fine-tuning on the domain from which the test data has been sampled from is often unfeasible. Even if the distribution from which the test data has been sampled from is accessible, labelling can be cumbersome and expensive. This issue is particularly relevant for video tasks, which often require per-frame pixel-wise labels (*e.g*. [9, 29]). Nonetheless, videos contain a wealth of information, especially if we assume that some (unlabelled) frames from the test distribution are available before actually performing the evaluation. Consider the practical case in which a drone for aerial photography is deployed in an unseen environment; a snowy weather for instance. It is then reasonable to assume that the first few seconds of its unlabelled video feed can be used to adapt its models to the surroundings in which it will soon be operating. Intuitively, collecting unlabelled sample videos from the new domain is a significantly simpler task than obtaining labelled data.

In this paper, we explore how unlabelled video data can be exploited in order to adapt pre-trained models to the distribution to which the test set belongs. Self-supervised methods are of particular interest for our scenario, as their objective allows for "fine-tuning without labels". In our evaluation, we address the task of video object segmentation, also known as dense tracking [6], as it has often been used to compare self-supervised methods trained on video data [42, 19, 15]. In this task, the pixel-wise mask of the target object is provided at test time in the first frame of the video, and the goal is to track the object of interest throughout the video sequence by providing per-frame masks.

Beside not having access to labelled data from the test distribution, in our experiments we consider another important condition: We assume to have received an already-trained model, and to not have access to neither its training routine nor to the data it has been trained with. This scenario is becoming increasingly important This scenario has recently become of great interest because of the increasingly prohibitive cost of training large-scale state-of-the-art models [4].

Recently, several works in the image domain have studied a (*de facto*) similar setup under the name of *test-time adaptation*. [26, 33, 37, 43]. However, these methods of-

ten rely on batch normalisation and implicitly assume the availability of batches with elements sampled i.i.d. from the test distribution to be used for adaptation. In contrast, when adapting the model with data originating from a video stream like in our case, this assumption is inevitably violated. In this paper, we re-purpose several test-time adaptation methods used in the image domain and experiment to which extent they can be useful with video data. In particular, we are interested in evaluating how well we can exploit unlabeled videos by using self-supervised objectives for improving the test performance on the downstream task. To this end, we investigate two distinct scenarios of arbitrary and severe domain shift. In the former case, we perform the test-time adaptations on unseen videos, whose originating distribution may differ from the distribution of the training data in arbitrary and unknown ways. In the latter, we impose severe (but controlled) domain shifts by artificially adding perturbations to the video frames.

In summary, the contributions of this paper are two-fold: First, we introduce a problem formulation to investigate the potential of using unlabeled video data for test-time adaptation in a self-supervised manner. Then, we perform an extensive evaluation to understand the behaviour of current state-of-the-art dense tracking methods in presence of several types of domain shifts and the impact of test-time adaptation in alleviating their detrimental effects.

2. Related Work

Unsupervised Domain Adaptation addresses a setup where the labeled data from a source domain and unlabeled data from a target domain are available during the training phase. The goal is to maximize the performance on the target domain [45]. In this regard, [36, 13, 18] propose feature alignment and adjusting the statistics of the source and target data distributions by applying linear transformations on the source features to lessen the impact of domain shift. Carlucci *et al*. [8] develop domain alignment layers that apply domain-specific operations and align the features from the source and target distributions to a reference and can be embedded in any network. Similarly, [28] introduces a moment matching component for multi-source domain adaptation, which is responsible for adapting the input domains to a target distribution. [38] employs a feature-wise transformation layer which learns the parameters of a linear operation and modulates the activations and adapt them to the target task/domain. In a different setup, Liang *et al*. [24] suggest an effective transfer learning approach in a scenario where a pre-trained model is to be adapted to a target domain without having access to the source data.

Domain Generalization considers a more general scenario where the target data distribution is unavailable during training [48]. The goal is to improve the performance on the target domain with a focus on enhancing the *training* process. In this respect, [7] proposes a multi-task setup and shows that training together with the auxiliary task of solving the jigsaw puzzle [27] improves the generalization to unseen domains. [23] proposes a meta-learning approach in which the objective for improving the generalization is learned itself, in contrast with methods that utilize manually designed loss functions [25, 3]. Several works have studied this aspect in an attempt to accustom the normalization layer to the target distribution [22, 26, 33, 34, 47, 5]. For example, [47] develops a domain-invariant normalization layer for stereo-matching by normalizing the features along the spatial and the channel dimensions to enforce the domain invariance in the learned representation while [34] proposes a domain-specific normalization layer for multi-source domain generalization by combining batch normalization [14] with instance normalization [39] where the combination weights are learnable parameters of the network.

Test-time Adaptation unlike the previously mentioned methods, performs domain generalization by learning from the data available at *test* time. In this respect, Sun *et al*. [37] propose a multi-task setup using supervised and self-supervised objectives where the auxiliary loss is used to further finetune the network during inference. In [43], the authors utilize entropy minimization [10, 35, 32, 31] to modify the modulation parameters of the BN layer to mitigate the impact of covariate shift between the training and testing data distributions and [26, 33, 22] suggest updating the normalization statistics of the BN layer as an effective way for adapting the features to the target domain. In this work, we build on test-time adaptation approaches, as we adapt the pre-trained models to the new unseen domains in video data. More details on our problem definition can be found in Section 3.1.

Dense Tracking also known as video object segmentation is the task of pixel-wise tracking a target object where the mask of the first object appearance is provided [6, 2, 1]. In recent years there has been a surge of interest in self-supervised methods for different applications [16], including correspondence matching and dense tracking. One of the earliest works in the deep learning-based methods for self-supervised correspondence learning was [42]. In this approach, the authors show that the feature embeddings learned by performing the auxiliary task of video colorization can be utilized for spatio-temporal correspondence matching via tracking matching features. Following works [20, 44] significantly enhance the performance of this method by adding several improvements such as cycle consistency, improved training procedure, using memory and attention mechanism. [21] combines region-level correspondence matching (tracking) with colorization while [46] and [17] utilize motion information for dense tracking. In a different line of work, [15] suggests a framework where the video is processed into a graph by dividing a frame

into multiple patches (nodes). They train the embeddings by performing a random walk on the constructed graph using a cycle consistency objective. In this paper, we use the current state-of-the-art methods VideoWalk [15] and MAST [19] as our baselines. Further details about these algorithms are provided in Section 3.2.

3. Problem Setup and Methods

This section discusses our proposed problem formulation and experimental setup, followed by an overview of the utilized baselines and test-time adaptation algorithms. Our primary focus lies on studying the impact of covariate shifts in the task of self-supervised dense tracking and the possible remedies utilizing the unlabeled video data. We are interested in studying ways to adapt a pre-trained model to the target data distribution without altering the training regime. This setup is beneficial due to the many practical use-cases in real-world conditions. Inspired by test-time adaptation literature from the image domain [37, 43, 26, 33], we explore utilizing unlabelled video data for addressing the problem of covariate shift in the video domain.

Test-time adaptation methods in the image domain usually assume the availability of a *diverse* batch of unlabeled data from the target distribution during inference. These data are used for further finetuning the model with an unsupervised or self-supervised objective. As a video contains much more information than a single image, in this work we study the extent in which the unlabeled video frames can be utilized for test-time adaptation. We note that the definition of *domain* in the literature is relatively imprecise. For example, it is unclear if we consider a dataset as a single domain or a combination of multiple domains (each class forming a cluster can be viewed as a separate domain). Hence, we initially contemplate a hypothesis where each video can be considered as an individual domain. Next, we enforce domain shift by manually adding various perturbations to the test videos [11]. To this end, we ask the following questions:

- Assuming each video represents a specific domain, how effective are the current test-time adaptation methods when applied to the task of dense tracking in videos?
- Considering the self-supervised setups for dense tracking, can further finetuning the model on the target video (essentially overfitting to a specific video domain using the self-supervised objective) improve the performance on the downstream task?
- In the case of clear domain shift such as noisy data, how effective are these adaptation methods for recovering the performance in self-supervised dense tracking tasks?

To answer to these questions, we experiment with modified variants of three recent approaches for test-time adaptation from the image domain, namely **Prediction-time BN** [33, 26], **TENT** [43], and **TTT** [37]. However, our setup is different from these methods as discussed in the following: First, the aforementioned methods are developed for image classification and assume that a diverse batch of data from the target distribution will be available at test time. In our setup, each video is considered as an individual domain, and the frames sampled from a single video comprise the batch, meaning the batch might not contain enough diversity. Unlike the prediction-time BN scenario in [26, 33], the captured statistics from a video sequence may not be diverse enough, so replacing the normalization statistics in the normalization layer with those collected from the video frames might hurt the performance. Therefore, we experiment with different momentum values as:

$$\hat{x} = (1 - \alpha) \times x_{old} + \alpha \times x_{new} \tag{1}$$

where x_{old} is the statistics estimated from the training data, x_{new} is the statistics computed from the video at hand, and $\alpha \in [0, 1]$ is the momentum. Second, these methods build on top of models trained in a supervised manner, while we examine baselines that are trained in a self-supervised fashion. Third, we use a modified version of TENT [43] where the self-supervised objective substitutes the entropy loss. TENT minimizes the entropy of the class prediction, while in dense tracking, the first mask is required for computing the pixel-wise label probabilities. As we aim at the fully unlabeled test-time adaptation, we utilize the self-supervised objective (eqs. (6) and (9)) instead of the entropy minimization. We refer to this adapted version as TENT*.

In our experimental setup, we consider two outlines for the offline and online applications. In offline applications such as video editing, all the video frames are available beforehand and can be exploited for adaptation. However, this is clearly not possible in online applications such as autonomous driving, as real-time inference is required. Nevertheless, it is still reasonable to assume that we have access to a limited amount of unlabeled data from the target domain. Therefore, we utilise all the video frames for test-time adaptation in the first scenario, and only use a fraction of frames in the latter. As our self-supervised dense tracking baselines, we chose two state-of-the-art methods of VideoWalk [15] and MAST [19]. In the following, we briefly explain the utilized test-time adaptation methods as well as the selected dense tracking baselines.

3.1. Test-time Adaptation

Prediction-time BN [26, 33] suggests replacing the statics of the normalization layer (μ, σ) with the ones estimated from the test data. [26, 33] observe that in a scenario where there is a domain shift between the training and testing data,

it is sub-optimal to normalize the activations with the μ and σ estimated from the training data. Assuming a batch of data from the target distribution is available at inference time, they propose to either replace [26] or update [33] the normalization statics with the ones computed from the test data.

TENT [43] algorithm proposes to update the normalization statistics as well as the shift and scale parameters γ and β in the BN layer and adapt the feature modulation to the target data distribution. In [43], the authors use entropy minimization as their optimization objective:

$$H(\hat{y}) = -\sum_{c} p(\hat{y}_c)\ log\ p(\hat{y}_c) \tag{2}$$

where $p(\hat{y}_c)$ is the network output probability for class c. As mentioned earlier, in our adopted variant of this method referred to as TENT*, we experiment with the self-supervised objectives (Equation 6 and Equation 9) instead of the entropy loss in Equation 2.

Test-time Training (TTT) [37] alters both the training and inference procedures. In [37], the authors modify the architecture to include a shared backbone as well as two separate heads for the main task (image classification) and a self-supervision objective, namely rotation classification. The model is then trained with the standard image classification objective together with the auxiliary loss in a multi-task setup. During the test phase, the model is further fine-tuned using the auxiliary objective. This way, the parameters of the shared backbone are modified and adapted for the target distribution, but the classification head remains unchanged. Therefore, the auxiliary head is utilized to adapt the backbone to the target data distribution and mitigate the impact of covariate shift between train and test data distributions.

3.2. Self-supervised Dense Tracking

Recently, self-supervised methods for dense tracking have significantly improved and achieve impressive performance, comparable to supervised counterparts [19, 15]. One of the earliest works in this area was [42], where the authors proposed to learn the correspondences based on video colorization. This algorithm has been the basis for many other approaches such as [20, 19]. In this method, a self-supervised objective for correspondence matching is defined based on colorizing the frames in a video. To this end, consider a colored reference frame where each pixel has a value $c_i \in R^d$ (colors are quantized to d bins) and a grayscale target image. The colors in the target frame are quantified as:

$$y_j = \sum_{i} A_{ij} c_i \tag{3}$$

where A is a similarity matrix computed as:

$$A_{ij} = \frac{exp(f_i^T f_j)}{\sum\limits_{k} exp(f_k^T f_j)} \tag{4}$$

In Equation 4, f represents the image features computed by a neural network which is trained by minimizing the following objective.

$$Loss = \sum_{j} CrossEntopy(y_j, c_j) \tag{5}$$

Here, c_j is the correct quantized color and y_j is the predicted color class by the model, based on Equation 3. During inference time, the learned features are utilized for various applications such as pose tracking and video object segmentation.

MAST [19] is one of the state-of-the-art methods based on colorization. In [19], the authors make several improvements to [42] such as explained in the following. First, they suggest using lab color space instead of RGB due to less correlation between color channels. Second, they enhance the architecture by incorporating a memory bank and employing an attention mechanism to retrieve the color in each target frame from multiple past frames using the attention weights. Third, they propose to replace the classification objective in Equation 5 with regressions as:

$$Loss = \frac{1}{n} \sum_{i} \begin{cases} 0.5(\hat{I}_t^i - I_t^i)^2 & \text{if } |\hat{I}_t^i - I_t^i| < 1 \\ |\hat{I}_t^i - I_t^i| - 0.5 & \text{otherwise} \end{cases} \tag{6}$$

where n is the number of pixels and $\hat{I}_t^i$ and I_t^i are the estimated and the actual color values for the i_{th} pixel, respectively. The intuition is that quantizing the colors to a limited number of classes causes loss of important information and leads to sub-optimal performance whereas using regression preserves all the color information.

VideoWalk [15] on the contrary, develops a framework for correspondence matching based on learning the patch-wise similarities across the video frames. In this algorithm, a space-time graph is formed by dividing each frame into multiple nodes (patches) and computing the edge weights based on a similarity metric between the neighboring nodes (across time and spatial dimensions). Consequently, the task of finding the correspondences across the video frames is devised as a contrastive random walk with patch-wise affinities providing the transition probabilities.

Assume q_t^i is the feature embedding of i_{th} node/patch at time-step t. The Affinity matrix between each two nodes in consecutive frames can be calculated as:

$$A_t^{t+1}(i,j) = \frac{exp(\langle q_t^i, q_{t+1}^j \rangle / \tau)}{\sum_{l=1}^{N} exp(\langle q_t^i, q_{t+1}^l \rangle / \tau)} \tag{7}$$

where τ is a temperature parameter. Subsequently, the long range affinity between the nodes from non-consecutive frames are computed as:

$$\hat{A}_t^{t+k} = \prod_{i=0}^{k-1} A_{t+i}^{t+i+1} = P(X_{t+k}|X_t) \tag{8}$$

The goal is to train the embeddings such that higher weights are assigned to the edge between the similar patches so that the random walk likely follows the path of the corresponding nodes. To achieve this goal, [15] uses an objective based on cycle-consistency and creates a palindrome of video frames so that for each node at the first time-step, we know the target node at the end of the walk. This objective can be formulated as:

$$Loss = CrossEntropy(\hat{A}_t^{t+k}, Y_t^{t+k}) \tag{9}$$

where Y_t^{t+k} is the actual corresponding node which is known as a result of the palindrome setup and the cycle consistency.

4. Experiments

4.1. Datasets

In this work, we experiment with DAVIS2017 [29] and TAO-VOS [40] datasets, two standard benchmarks for evaluating dense tracking methods.

DAVIS2017 [29] validation set consists of 30 videos with an average duration of 3.4 seconds. As the videos in DAVIS are somewhat short, especially for the online setup where half of the video frames are used for adaptation, we further benchmark this setup on a sub-set of **TAO-VOS** [41, 9]. The videos in this dataset are considerably longer than DAVIS2017 with an average length of 36.7 seconds. Therefore even when using half of the frames for the evaluation, we end up with longer videos than DAVIS2017. We selected this subset based on two criteria: each video contains at least 1000 frames and at most 2 target objects. The latter condition is a practical consideration as the current self-supervised methods do not work well in scenes with many objects. Therefore, we resort to relatively more straightforward videos with more frames, allowing us to use a subset of data for test-time adaptations. The full list of the selected videos can be found in the supplementary material. We report the standard evaluation metrics of dense tracking task, *Region Similarity* and *Contour Accuracy* ($J\&F$) scores [29]. J refers to the intersection-over-union between the model prediction and the ground-truth and F measures the quality of the estimated object boundaries.

4.2. Results

In this section, we present our experimental results following the setup explained in Section 3 and analyze the observed behavior.

Table 1 presents the results for offline and online setups on DAVIS2017 dataset. Each row shows the J and F scores of the baselines in the presence of a specific domain shift, with and without test-time adaptation. In the first block of rows, we investigate the efficacy of test-time adaptation with a self-supervised objective on the test data with arbitrary domain shifts (without any added perturbation). As we are working with self-supervised baselines, a question naturally arises whether further tuning on a specific video is helpful and to which extent it can improve the performance on the downstream task. Next, we study a scenario with a substantial domain shift between the training and testing data distributions. In this respect, we follow the proposed setup in [12] and impose an artificial covariate shift to the video frames. In particular, we experiment with Gaussian noise, Motion Blur, Fog, and Snow perturbations, shown in Figure 1. Perturbations are generated according to the level 5 severity as described in [11].

Figure 1: Samples from corrupted data distributions (Gaussian noise and Snow at the top row, Fog and Motion Blur at the bottom row).

As can be seen from the results in table 1, self-supervised test-time adaptation on the data without perturbation slightly improves the results, while considerably decreasing the adverse effect of covariate shift for data with severe perturbations. The behavior in arbitrary domain shift scenario (without perturbation) implies that in situations with mild distribution shift, overfitting to the current self-supervised objectives does not fully transfer to the downstream task and only marginally improves the performance. However, these methods can successfully adapt the features to the target domain when there is a severe distribution shift between the training and testing data. Interestingly, in most cases, updating the normalization statics (BN column) has an equal or superior positive impact on the dense tracking accuracy despite its simplicity. However, we note that Fog perturbation is an exception where both MAST and VideoWalk methods achieve considerably better accuracy with TENT* and TTT algorithms. Furthermore, the results show a similar pattern in offline and online scenarios, suggesting

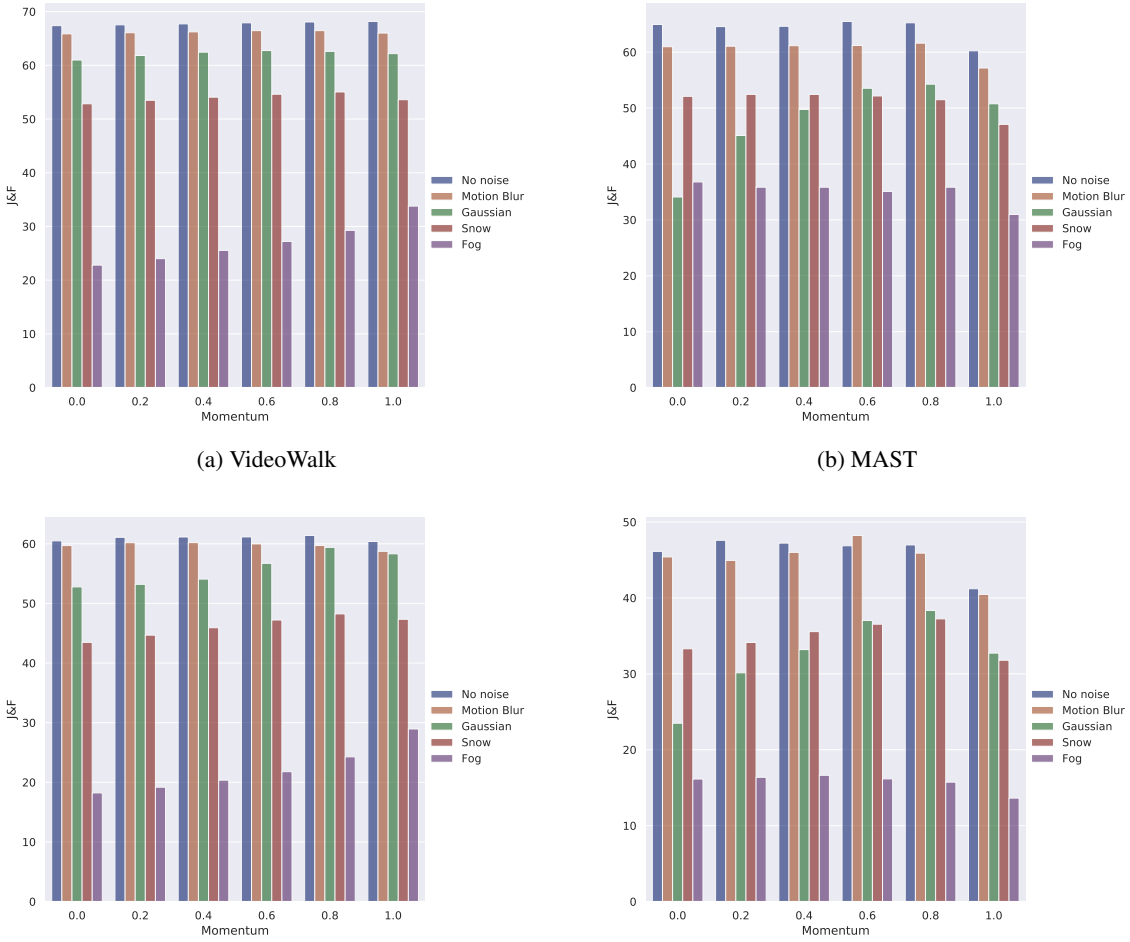
Dense Tracking (Offline)				Dense Tracking (Online)				Test-time Adaptation			
VideoWalk		MAST		VideoWalk		MAST		BN	TENT*	TTT	Noise
J	F	J	F	J	F	J	F				
64.38	*70.40*	*62.95*	*66.94*	*69.46*	*74.43*	*67.11*	*70.85*				—
+1.00	**+0.56**	**+0.47**	+0.62	+0.67	**+0.99**	**+1.04**	**+1.04**	✓			
+1.04	+0.50	+0.32	**+0.65**	**+0.70**	+0.97	+0.20	+0.30		✓		
+1.17	+0.47	+0.09	+0.34	+0.64	+0.84	+0.27	+0.39			✓	
58.40	*63.08*	*32.70*	*35.48*	*64.43*	*67.89*	*41.51*	*43.36*				Gaussian
+1.85	+2.16	**+19.82**	**+20.54**	+2.07	+2.58	**+18.21**	**+19.26**	✓			
+1.91	+2.44	+17.98	+18.77	**+3.73**	**+3.91**	+15.90	+17.17		✓		
+2.67	**+2.97**	+18.06	+18.15	+2.11	+2.20	+15.37	+16.58			✓	
62.97	*68.75*	*58.49*	*63.45*	*67.69*	*72.50*	*64.54*	*69.99*				Motion Blur
+0.69	**+0.51**	**+0.49**	**+0.80**	+1.01	+1.62	**+0.35**	**+0.10**	✓			
+0.41	+0.34	-0.10	+0.13	**+1.04**	**+1.69**	-0.21	-0.22		✓		
+0.18	+0.11	+0.12	-0.18	+0.97	+1.28	-0.58	-0.43			✓	
50.89	*54.77*	*51.12*	*53.08*	*56.44*	*59.20*	*58.51*	*59.68*				Snow
+1.63	+2.78	**+0.83**	**+0.77**	**+2.60**	**+2.80**	+0.51	+0.46	✓			
+1.99	+2.80	+0.14	+0.34	+2.43	+2.52	**+0.77**	**+0.99**		✓		
+2.79	**+3.92**	+0.32	+0.39	+1.98	+1.91	+0.15	+0.38			✓	
19.27	*26.32*	*35.55*	*38.05*	*24.76*	*30.76*	*43.42*	*45.03*				Fog
+11.23	+10.76	0.00	0.00	+11.54	+9.860	0.00	0.00	✓			
+12.01	+12.23	+3.09	+2.66	+9.67	+9.22	+3.83	+3.51		✓		
+18.70	**+18.42**	**+9.85**	**+8.50**	**+14.07**	**+14.21**	**+9.24**	**+9.54**			✓	

Table 1: J and F scores for VideoWalk [15] and MAST [19] self-supervised dense tracking methods on **DAVIS2017** validation set in **offline** and **online** settings. In the offline mode, all video frames are used for adaptation. In the online setup, we use the first and second half of the video for adaptation and evaluation, respectively. For each perturbation variant, we compare the accuracy of the baseline model with three test-time adaptation techniques as explained in Section 3. Results in cursive correspond to absolute metrics, followed by their delta when using one of the test-time adaptation methods. Best results per column shown in bold.

that performing test-time adaptation is beneficial for both circumstances.

For the results shown in column BN, we experimented with different momentum values and updated the normalization statistics according to Equation 1. Here the results are provided with the best-found momentum, and additional results can be seen in Figure 2. From these plots, we see that partially updating the normalization statistics with those from the target domain alleviates the impact of covariate shift, but completely replacing them (momentum value of 1) can deteriorate the performance. This behavior can be due to a lack of diversity in video frames, resulting in sub-optimal performance when ignoring the information collected from the training data (the old normalization statistics). Moreover, we observe varying trends in the VideoWalk and MAST methods; for example, in Fog perturbation, VideoWalk enjoys updating the normalization statistics, whereas it is better to keep the statistics unchanged for MAST. This can result from different training objectives in these approaches as the self-supervised loss in MAST is purely based on color information (Equation 6), while VideoWalk additionally utilizes higher-level correspondences between pixel embeddings (Equation 9).

As explained in Section 3.1, TTT [37] and TENT [43] approaches finetune the network weights. We note that updating the model weights also depends on how the normalization statistics in the BN layer are handled (i.e., training the model when freezing or updating the BN statistics). In these methods, it is assumed that a diverse batch of data is available, but this condition may not hold when sampling the batch from a video sequence. Therefore, we need to consider this factor and carefully treat the BN layer. We experimented with both cases of training with freezing and updating the normalization statistics. The results in Tables 1 and 2, are with the best-found configuration an additional results in this regard can be found in the supplementary material. Concerning the other hyperparameters in VideoWalk and MAST, we performed an extensive hyperparam-



(a) VideoWalk

(b) MAST

(c) VideoWalk

(d) MAST

Figure 2: Ablation on the momentum in prediction-time BN (Equation 1) and the impact it has on the performance of VideoWalk and MAST methods under different perturbations. The first and second rows illustrate the results on DAVIS and TAO-VOS datasets, respectively. The diagrams on DAVIS are from the offline setup (we observed a similar trend in the online mode). The results indicate that, except for Fog, it is better to update the normalization statics with a momentum value less than one in most cases. In VideoWalk, it is better to completely replace the statistics with those collected from the target domain, while in MAST, it is better to keep the statistics unchanged.

eter search and found that the parameters used during the self-supervision also worked best for the test-time adaptation. We utilized the public available code from the respective authors and further finetuned the model until the convergence of the self-supervised loss.

Considering the short duration of DAVIS2017 videos, utilizing half of the video may not provide solid conclusions. Therefore, we also benchmark the baselines on a subset of TAO-VOS, which contains about ten times longer videos than DAVIS2017, with a similar experimental setup described before. Table 2 presents the results for this dataset in online setting where we use the first half of the video for adaptation and the rest for evaluation. Furthermore, figs. 2c and 2d show an ablation on the performance of the baselines when updating the normalization statistics with varying momentum values, as in Equation 1. The results in BN column in Table 2 are obtained using the best-found momentum based on this ablation. We observe a similar pattern between the results obtained from DAVIS2017 and the longer videos in the TAO-VOS subset, validating the efficacy of test-time adaptation for short and long videos.

5. Conclusion

In this work, we investigated the role that self-supervision can have in alleviating the harmful effect of distribution mismatch between train and test datasets of video data. We considered two scenarios of practical relevance.

Dense Tracking				Test-time Adaptation			
VideoWalk		MAST		BN	TENT*	TTT	Noise
J	F	J	F				
55.83	*65.17*	*43.68*	*48.59*				—
+0.83	+0.98	**+1.49**	+1.42	✓			
+1.23	+1.52	+1.21	+1.46		✓		
+1.42	**+1.76**	+0.85	**+2.11**			✓	
48.29	*57.25*	*22.34*	*24.64*				Gaussian
+6.51	**+6.78**	**+14.40**	+15.31	✓			
+3.56	+3.90	+13.79	+14.71		✓		
+4.35	+4.56	+14.21	**+15.33**			✓	
55.21	*64.19*	*42.69*	*48.12*				Motion Blur
+0.71	+0.29	**+2.71**	**+2.95**	✓			
+1.01	**+1.09**	+2.32	+1.93		✓		
+0.58	+0.43	+2.65	+2.71			✓	
38.79	*48.16*	*31.53*	*35.07*				Snow
+5.18	+4.35	**+3.94**	+3.95	✓			
+6.48	**+5.88**	+2.82	+2.81		✓		
+6.56	+5.31	+3.87	**+4.11**			✓	
14.47	*21.96*	*14.60*	*17.67*				Fog
+11.11	+10.07	+0.52	+0.49	✓			
+23.64	**+24.62**	+1.63	+2.10		✓		
+22.24	+22.20	**+6.12**	**+5.48**			✓	

Table 2: J and F scores for VideoWalk [15] and MAST [19] self-supervised methods on a subset of the **TOA-VOS** dataset in an **online setup** when using half of the video for adaptation and evaluation on the second half of the frames. Results in cursive correspond to absolute metrics, followed by their delta when using one of the test-time adaptation methods. Best results per column shown in bold.

One, for offline applications, in which the entire video sequence is available in advance. Another, for online applications, in which instead we are interested in real-time inference and have access to some unlabeled data from the target domain prior to inference. In both cases, we only considered pre-trained model and we assume to not have access to neither their training data nor training routine. We studied the behavior of two recent self-supervised dense tracking algorithms in the presence of several domain shifts. Our experimental results confirm that self-supervised test-time adaptation is an effective method for decreasing the impact of covariate shift in dense tracking, but that the extent of its efficacy largely depend on the specific shifts and algorithms in question.

6. Acknowledgement

This work was supported by the TU Kaiserslautern CS PhD scholarship program, the BMBF project ExplAINN (01IS19074), and the NVIDIA AI Lab (NVAIL) program. Further, we thank all members of the Deep Learning Competence Center at the DFKI for their feedback and support.

References

[1] Fatemeh Azimi, Benjamin Bischke, Sebastian Palacio, Federico Raue, Jörn Hees, and Andreas Dengel. Revisiting sequence-to-sequence video object segmentation with multi-task loss and skip-memory. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5376–5383. IEEE, 2021.

[2] Fatemeh Azimi, Stanislav Frolov, Federico Raue, Joern Hees, and Andreas Dengel. Hybrid-s2s: Video object segmentation with recurrent networks and correspondence matching. *arXiv preprint arXiv:2010.05069*, 2020.

[3] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems*, 31:998–1008, 2018.

[4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[5] Collin Burns and Jacob Steinhardt. Limitations of post-hoc feature alignment for robustness. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2525–2533, 2021.

[6] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

[7] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[8] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. Autodial: Automatic domain alignment layers. In *2017 IEEE international conference on computer vision (ICCV)*, pages 5077–5085. IEEE, 2017.

[9] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. *arXiv preprint arXiv:2005.10356*, 2020.

[10] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

[11] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *iclr*, 2019.

[12] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[13] Haoshuo Huang, Qixing Huang, and Philipp Krahenbuhl. Domain transfer through deep activation matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 590–605, 2018.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[15] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*, 2020.

[16] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[17] Shu Kong and Charless Fowlkes. Multigrid predictive filter flow for unsupervised learning on videos. *arXiv preprint arXiv:1904.01693*, 2019.

[18] Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, William T Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. *arXiv preprint arXiv:1811.05443*, 2018.

[19] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2020.

[20] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. *arXiv preprint arXiv:1905.00875*, 2019.

[21] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. *arXiv preprint arXiv:1909.11895*, 2019.

[22] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[23] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, pages 3915–3924. PMLR, 2019.

[24] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[25] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013.

[26] 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.

[27] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[28] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019.

[29] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[30] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

[31] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9471–9480, 2019.

[32] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019.

[33] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving

robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33, 2020.

[34] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 68–83. Springer, 2020.

[35] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[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.

[37] 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 *icml*, 2020.

[38] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. *arXiv preprint arXiv:2001.08735*, 2020.

[39] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[40] Paul Voigtlaender, Lishu Luo, Chun Yuan, Yong Jiang, and Bastian Leibe. Reducing the annotation effort for video object segmentation datasets. *arXiv preprint arXiv:2011.01142*, 2020.

[41] Paul Voigtlaender, Lishu Luo, Chun Yuan, Yong Jiang, and Bastian Leibe. Reducing the annotation effort for video object segmentation datasets. In *WACV*, 2021.

[42] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 391–408, 2018.

[43] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[44] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

[45] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

[46] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, 2021.

[47] Feihu Zhang, Xiaojuan Qi, Ruigang Yang, Victor Prisacariu, Benjamin Wah, and Philip Torr. Domain-invariant stereo matching networks. In *European Conference on Computer Vision*, pages 420–439. Springer, 2020.

[48] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *arXiv preprint arXiv:2103.02503*, 2021.