GENERATIVE TARGET UPDATE FOR ADAPTIVE SIAMESE TRACKING

Madhu Kiran[‡]   Le Thanh Nguyen-Meidine[*]   Rajat Sahay[‡]   Rafael Menelau Oliveira E Cruz[*]
Louis-Antoine Blais-Morin[§]   Eric Granger[*]

February 22, 2022

ABSTRACT

Siamese trackers perform similarity matching with templates (i.e., target models) to recursively localize objects within a search region. Several strategies have been proposed in the literature to update a template based on the tracker output, typically extracted from the target search region in the current frame, and thereby mitigate the effects of target drift. However, this may lead to corrupted templates, limiting the potential benefits of a template update strategy. This paper proposes a model adaptation method for Siamese trackers that uses a generative model to produce a synthetic template from the object search regions of several previous frames, rather than directly using the tracker output. Since the search region encompasses the target, attention from the search region is used for robust model adaptation. In particular, our approach relies on an auto-encoder trained through adversarial learning to detect changes in a target object's appearance, and predict a future target template, using a set of target templates localized from tracker outputs at previous frames. To prevent template corruption during the update, the proposed tracker also performs change detection using the generative model to suspend updates until the tracker stabilizes, and robust matching can resume through dynamic template fusion. Extensive experiments conducted on VOT-16, VOT-17, OTB-50, and OTB-100 datasets highlight the effectiveness of our method, along with the impact of its key components. Results indicate that our proposed approach can outperform state-of-art trackers, and its overall robustness allows tracking for a longer time before failure.
Code: https://anonymous.4open.science/r/AdaptiveSiamese-CE78/

1 Introduction

Many video analytics, monitoring, and surveillance applications rely on visual object tracking (VOT) to locate targets appearing in a camera viewpoint over time, scene understanding, action and event recognition, video summarizing, person re-identification. In real-world video surveillance applications, VOT is challenging due to real-time computational constraints, changes and deformation in target appearance, rapid motions, occlusion, motion blur, and complex backgrounds. In real-time video surveillance applications, the time required to capture and identify various events is a significant constraint. Techniques for VOT may be categorized according to the target model or template construction mechanism, as either generative or discriminative. Generative appearance models represent target appearance without considering the background, while discriminative trackers learn a representation to distinguish between a target and background Salti et al. [2012]. The trackers can be further classified based on their image representation techniques, ranging from conventional hand-crafted descriptors Hare et al. [2016], Henriques et al. [2015], Nebehay and Pflugfelder, Wang et al. to more recent deep learning models, like Siamese trackers Bertinetto et al. [2016], Guo et al. [a], Li et al. [a], Li and Zhang [2019], Zhang and Peng, Zhang et al. [a], Zhu et al..

---
[*]Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA), Ecole de technologie superieure, Montreal, Canada
[†]Corresponding author, madhu_sajc@hotmail.com
[‡]Vellore Institute of Technology, Vellore
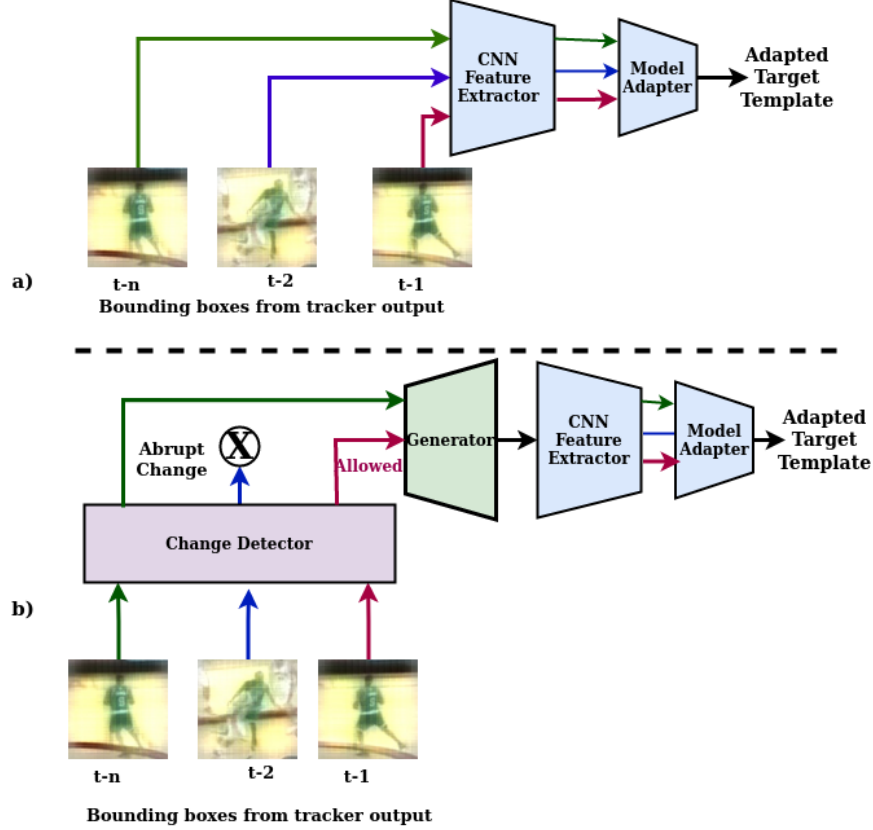[§]Genetec Inc.

Figure 1: Approaches to select templates for adaptive Siamese tracking. (a) Conventional approaches select templates from previous tracker outputs. (b) Our approach generates templates from previous ones using a generative model, and filters noisy templates via change detection.

One of the initial Siamese trackers – the Fully Convolutional Siamese tracker (SiameseFC) Bertinetto et al. [2016] – uses a single features representation extracted at the beginning of a trajectory, and does not update the target features during tracking. Although this strategy can provide computational efficiency, SiameseFC trackers suffer from target drift over time. Target drift is defined as a situation when the tracker slowly starts focusing on distractive backgrounds (rather than the target), and eventually looses the target. Such drift causes broken tracklets, a potential problem in video surveillance applications such as loitering detection, video person re-identification, face recognition, and other related applications. When the object's appearance changes abruptly, or the object is occluded or partially leaves the search region, the SiameseFC tracker temporarily drifts to a location with a high response map score Zhang et al. [b]. Some adaptive Siamese trackers have been proposed that allow for template updates. Most early trackers sought to update target features as a moving average based on localizations from the current frame output. Other trackers apply strategies to address drifting objects by storing an array of templates, or combining features from various tracker outputs Yang and Chan, Zhang et al. [b]. However, these trackers face issues when updating templates on every frame based on tracker output. In particular, they integrate noise from the tracker output templates, especially in the present image occlusion or drift. Moreover, when training a Siamese tracker for matching based on multiple templates, learning the template update function in addition to the conventional search-template pair may lead to over-fitting Zhang et al. [b]. Hence, to avoid corrupting templates, it is important to manage when and how the templates are updated.

In this paper, we focus robust VOT of a single object, where the template is updated dynamically in response to changes in the object's appearance. This paper introduces a method that applied to any adaptive Siamese trackers for real-time applications. Instead of using the samples mined directly from the tracker output, we propose to use a generative model to generate a sample observing many previous target template. This generative model predicts the future appearance of a target template given a set of consecutive target templates localized from tracker outputs at previous frames. It also allows detecting abrupt changes in the appearance of target objects, and thereby preventing template corruption by suspending template updates until the tracker stabilizes. In the absence of an abrupt change, our generative model

outputs a synthetic target template for robust matching through dynamic template fusion, and updating the target template.

In contrast with Zhang et al. [b], our method learns the target update itself, using cross-attention between search region and template features. This allows selecting channels among the target features that are most useful for target update. The cross-attention approach relies on attention from the target's current appearance in the search region to update the existing target template. The proposed generative model is designed by adversarial training a video autoencoder to produce a future frame. The discrepancy between the generated future frame, and the target's appearance from tracker output helps detect appearance changes using a change detection mechanism. We summarise our contribution as follows. We propose a method for adaptation of Siamese trackers based generative model update. The generative model produces a future template by observing the the past templates. Additionally, change detection is proposed using the generative model to suspend model update during target drifting. Finally, the method relies on the difference between a simple average and a learned fusion templates to define an inequality constraint during learning of model adaptation. It uses attention from the search region to attend to salient regions in the tracker localised template. For proof-of-concept validation, the proposed method is integrated into state-of-art SiamFC+ and SiamRPN trackers Zhang and Peng, Li et al. [a], and compared to different conventional and state-of-art trackers from deep Siamese family Bertinetto et al. [2016], Zhang and Peng on videos from the OTB Wu et al. and VOT Kristan and et al., Kristan and et al. [2018] evaluations datasets. We also perform ablation studies on different modules to study the effectiveness of the proposed method.

2 Related Work

Pioneered by SINT Tao et al. and SiamFC Bertinetto et al. [2016], the Siamese family of trackers evolved from Siamese networks trained offline with similarity metrics. These networks were trained on a large dataset to learn generic features for object tracking. SiamRPN Li et al. [a] further improves on this work by employing region proposals to produce a target-specific anchor-based detector. Then, the following Siamese trackers mainly involved designing more powerful backbones Zhang and Peng, Li and Zhang [2019] or proposal networks, like in Fan and Ling. ATOM Danelljan et al. [a] and DIMP Bhat et al. [2019] are robust online trackers that differ from the general offline Siamese trackers by their ability to update model online during tracking. Other paradigms of Siamese trackers are distractor-aware training, domain-specific tracking He et al., Zhu et al..

In Zhong et al. [2018], an LSTM is incorporated to learn long-term relationships during tracking and turns the VOT problem into a consecutive decision-making process of selecting the best model for tracking via reinforcement learning Duman and Erdem [2019]. In Valmadre et al. and Zhu et al., models are updated online by a moving average based learning. These methods integrate the target region extracted from tracker output into the initial target. In Song et al., a generative model is learned via adversarial learning to generate random masks that produce shifted versions of target templates from the original template. Then, an adversarial function is used to decide whether or not the generated template is from the same distribution and if they will be used as templates for tracking. In Yang and Chan, an LSTM is employed to estimate the current template by storing previous templates in a memory bank. In Guo et al. [b], authors propose to compute transformation matrix with reference to the initial template, with a regularised linear regression in the Fourier domain. Finally, in Yao et al., authors propose to learn the updating co-efficient of a correlation filter-based tracker using SGD online. All these methods use the tracker output as the reference template while updating on top of the initial template. Bhat et al. [2019], Danelljan et al. [b] propose a model where an adaptive discriminative model is generated online by the steepest gradient descent method. They differ from another online learned method like Nam and Han due to their real-time performance. Similarly Zhang et al. [a] introduce online model prediction but employ a fast conjugate gradient algorithm for model prediction. Foreground score maps are estimated online, and the classification branch is combined by weighted addition.

Several methods follow the standard strategy of updating target template features, such as simple averaging, where the template is updated as a running average with exponentially decaying weights over time. This yields a template update defined by:

$$\widetilde{\varphi}^n = (1 - \gamma)\widetilde{\varphi}^{n-1} + \gamma\varphi^n, \tag{1}$$

where n denotes the time step, $\widetilde{\varphi}^n$ the predicted template, and γ the learning rate.

This strategy has several issues, most notably the possibility of integrating noise and corruption into templates during the update process. Therefore, authors in Zhang et al. [b] proposed a network which, when given an input of past template features, the template extracted from current tracker output produces a new representation that can be added to the original ground truth template (obtained during tracker initialization). This approach further suffers from the following issues. (1) A future template for the tracker is unseen at the time of template update, and the model is updated solely based on the tracker output in the past frame output. (2) The model is updated every frame making it still susceptible to the integration of noise over time. (3) Network training is a tedious task since it must be trained continuously offline by

running the tracker on the training dataset. It must produce a previous frame feature representation that needs to be stored and used for the next training iteration. Further developments in this direction are challenging.

3 Proposed Adaptive Siamese Tracker

Given the initial object location, a ground truth-object template image T is extracted, along with the corresponding deep CNN features φ_{gt}. A tracker seeks to produce object localization $BBox$ at a given time step by matching φ_{gt} with search region features φ_s. The objective is to produce a trajectory by recursively extracting search regions from tracker output, and matching them with a given template over each input video frame.

a) Template Prediction and Change Detection: Inspired from Tang et al. [2020], We employ a video autoencoder that is trained through adversarial learning for template generation. Given an set of past templates T^n where $n = t, t-1, t-2, t-3...$ we aim to predict a future template for time step t. As described below, our template generation method consists of a generator and a discriminator.

Generator: It consists of an encoder-decoder architecture (see Fig 2). The encoder compresses an input video clip into a small bottleneck with a set of CNN layers and Conv-LSTM based recurrent network to model the temporal relationship between the frames. The decoder consists of some layers of transposed CNN to obtain the predicted video frame. Hence given an input video clip of $T^{t-k}, ..., T^{t-2}, T^{t-1}$, the Generator produces the estimated future video frame $\hat{T}$. The generator is trained according to the Mean Squared Error (MSE) between predicted image $\hat{T}$ and ground truth image T

Discriminator: It comprises of several CNN layers to compete with the generator to differentiate between the ground truth and generated frames. The discriminator distinguishes a real-world image from a fake image, promoting the generator to produce good quality images. Since training the autoencoder on MSE loss alone will cause the output to be blurry, we leverage the discriminator to help produce higher-quality images. The labels are set to 0 for fake images (obtained from the autoencoder's reconstruction) and 1 for real (ground truth template image). The discriminator is
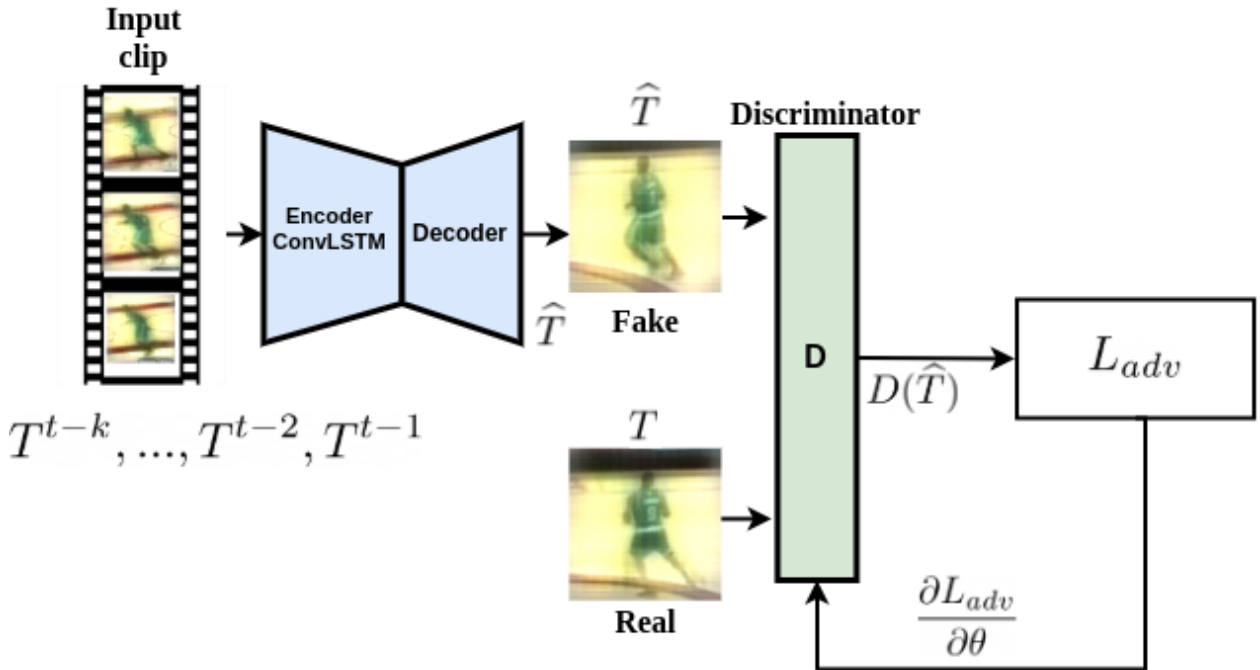

Figure 2: Our generator model is a video autoencoder that is trained adversarially. A future target template is reconstructed from a sequence of input target templates. The discriminator D processed the reconstructed template as fake, and the ground truth template input as real.
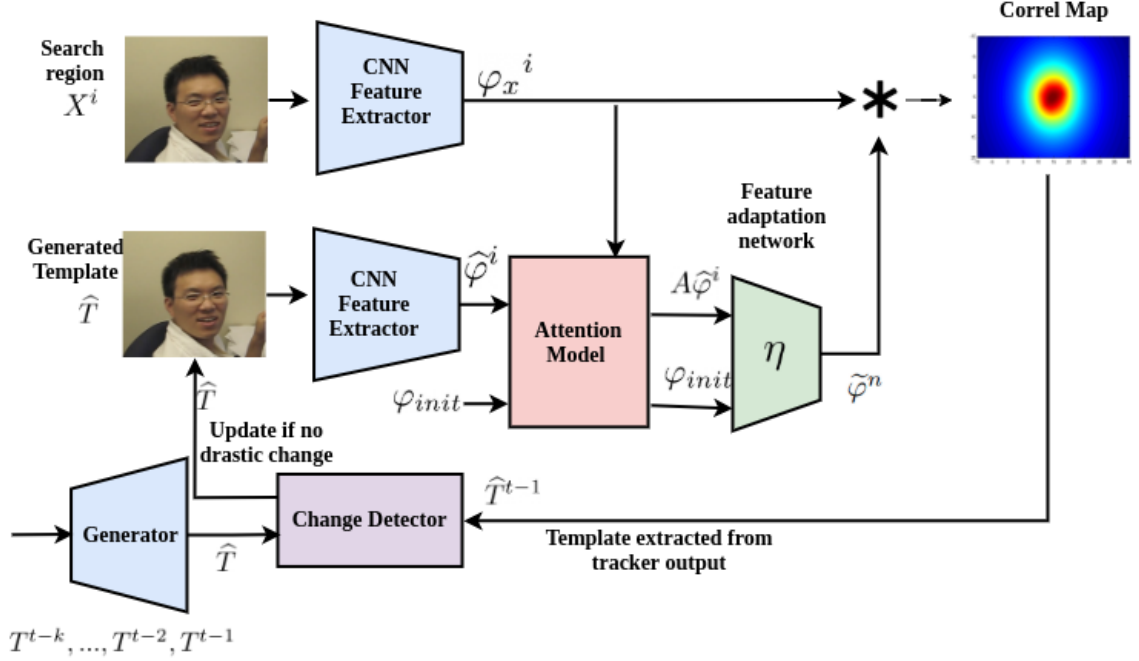
Figure 3: Block diagram of our proposed generic template update system for Siamese trackers that adapts the model of the target template with a generative model and change detection. Our attention based dynamic ensemble of targets adapts the model to the current representation of the target with attention from search region. The change detection system disables template update during anomalies such as occlusion and severe target drift.

trained with an adversarial loss:

$$L_{adv}^{D}(T, \widehat{T}) = \frac{1}{2}(D(T) - 1)^2 + \frac{1}{2}(D(\widehat{T}) - 0)^2 \qquad (2)$$

Change Detection: Once the adversarial auto-encoder has been trained, the average MSE error between the reconstructed template and search regions from each input frame in the video clip is computed to produce the reconstruction error. Similar to previous methods such as Duman and Erdem [2019], Zhao et al., we adopt the regularity score to detect abrupt changes in template clips. Let $e(T)$ be the reconstruction error. Reconstruction error should be normalized from the sequences of the same video with:

$$s(x) = 1 - \frac{e(T) - \min_T e(T)}{\max_T e(T)} \qquad (3)$$

In practice it is difficult to set $\min_T e(T)$ and $\max_T e(T)$ as the future frames are not observable. Hence, we set $\min_T e(T)$ and $\max_T e(T)$ experimentally using a validation set. The regularity score $s(x)$ serves as the measure for regular templates. Hence a score of less than a threshold τ is considered an abrupt change. The length of the input template sequence is kept fixed, and new templates are updated into the sequence by pushing the oldest template out of the stack. When a change is detected, the template that was last pushed into the stack is rejected and considered a possible source of corruption, and the template update eventually stalled for that particular time step.

b) Template Update with Cross Attention: Target model adaptation is often based on the last known appearance of the object from previous frames. At the start of the tracking, the initial target feature φ_{init} needs to be adapted to match the latest object appearance. Such adaptation is not possible without predicting the tracker in the current frame. At the same time, it is to be noted that the search region encompasses the target in the current frame, given that the change detector has detected no drastic change. Therefore we propose to use this cue to obtain attention from the search region to adapt the model. In addition to this, search region features and template features are of different sizes. This difference in feature size has inspired our proposal of using channel attention across the search and template stream. We follow a similar model adaptation paradigm as Zhang et al. [b] along with our attention model and proposed optimization with inequality constraints.Zhang et al. [b] consider adapting the target feature by adding additional information to the initial target feature φ_{init}.

Let $\widehat{\varphi}^t$ be the feature extracted from the generated template $\widehat{T}^t$. The generated template is the predicted target appearance. In comparison to $\widehat{\varphi}^t$, φ_{init} is the most reliable target feature as it is obtained during the initialization of the tracker using ground truth information. The model adaptation mechanism considers both φ_{init} and $\widehat{\varphi}^t$ to predict the adapted feature $\widetilde{\varphi}^t$. As discussed earlier, the first step is to obtain attention from the search region to select important channels in $\widehat{\varphi}^t$. Let $\varphi_z{}^t$ be the feature extracted from search region. Then we obtain matching channel attention from $\varphi_z{}^t$, φ_{init} and $\widehat{\varphi}^t$ by passing through an attention model similar to channel attention in Subramaniam et al. , using an MLP with Sigmoid activation to select channels based on importance. The attention obtained from $\varphi_z{}^t$, φ_{init} and $\widehat{\varphi}^t$ are averaged to obtain channel attention A. Attention A is multiplied with $\widehat{\varphi}^t$. Therefore the channels of $\widehat{\varphi}^t$ have been re-weighed and common saliency across search region and target template are encompassed into the attention.

The attended feature $A\widehat{\varphi}^t$ and $\widetilde{\varphi}^{t-1}$ (obtained from the prior frame after model adaptation) are then concatenating in the channel dimension as follows 4:

$$\varphi_{concat} = [\widetilde{\varphi}^{t-1}; A\widehat{\varphi}^t] \tag{4}$$

The concatenated feature φ_{concat} is passed through a two layer CNN with 1x1 convolution layer, followed by a TanH activation function to obtain adapted feature in:

$$\widetilde{\varphi}^t = \eta(\varphi_{concat}), \tag{5}$$

where η is the model adaptation network discussed above, and $\widetilde{\varphi}^t$ is the adapted target template for tracking.

c) Model Adaptation: During training, target samples are generated from the training data keeping the chronological order of the image frame in a video to obtain features $\varphi_{init}, \varphi_{GT}$. The ground truth video data generated these two, i.e., initial and template from future frames. To obtain the generated template, n consecutive templates are used from the same video to generate $\widehat{\varphi}^t$ by using the pre-trained generator that was previously discussed. To enable the system, learn to generate an adapted feature to resemble a target template from the next frame, we employ MSE loss:

$$L_{mdl-mse} = \|\varphi_{GT} - \widetilde{\varphi}^t\|_2, \tag{6}$$

where φ_{GT} are the ground truth target features which are chronologically the latest template. We expect the adapted template $\widetilde{\varphi}^t$ obtained by adapting previously seen target templates to resemble the future ground truth template.

Optimizing the MSE loss in our case is a difficult task since the model is being forced to learn to produce an unseen representation from future frames given two different previously seen frames. In Zhang et al. [b], the tracker is recursively train on several training cycles, which is a tedious task. Template update can also be performed by simply averaging features that would suffer from noisy updates and feature smoothing due to averaging both leading to information loss. Such simple averaging can be used as a cue to introduce a constraint to optimize the template update.

Let φ_{avg} be the averaged template obtained by averaging φ_{init} and φ^{t-1}. Let D_E denote the Euclidean distance function. It is reasonable to assume that simple template averaging is a trivial solution and therefore the distance between learnt template $\widetilde{\varphi}^t$ and φ_{GT}(the future template) must be less than φ_{avg} and φ_{GT}. Constrained loss given by,

$$L_{const-mse} = \|\varphi_{GT} - \widetilde{\varphi}^t\|_2 + \lambda\, ReLU((D_E(\varphi_{GT}, \widetilde{\varphi}) - D_E(\varphi_{GT}, \varphi_{avg})) \tag{7}$$

where ReLU ensures that the gradients are passed for the constraint only when the constraint is not respected. λ is set to a value $\gg 1$ and is determined experimentally.

4   Results and Discussion

a) Experimental Methodology: A ResNet-22 CNN similar to SiamDW tracker Sosnovik et al., Zhang and Peng is used for a fair comparison. The system on GOT-10K dataset Huang et al. [2019] to train our video autoencoder, as well as the tracking network similar to Sosnovik et al. for direct comparison since they use a similar baseline as ours. GOT-10K has around 10,000 video sequences with 1.5 million labeled bounding boxes to train our tracking network and auto encoder. In particular, due to many training sequences, the autoencoder overall motion model for objects in generic videos to predict frames in the future. We used the official training set of GOT10-K to train the networks. We use the same data augmentation techniques as Sosnovik et al., Zhang and Peng. The autoencoder was pre-trained adversarially with the discriminator. The Siamese tracker is pre-trained without the autoencoder by selecting random samples in a specific video, one for the template and the other for the search region.

The standard tracking benchmarks, OTB2013, OTB2015 Wu et al. and VOT2017 Kristan and et al. video datasets, are uses to evaluate trackers. The OTB Wu et al. dataset consists of sets OTB213 and OTB2015 with 50 and 100 real-world tracking videos, respectively. The metrics used with OTB datasets are success rate and precision. VOT2017 dataset has 60 public test videos with a total of 21,356 frames. The VOT protocol re-initializes the tracker when the tracker fails with a delay of 5 frames. Evaluation measures used with VOT are EAO and (Expected average overlap), a combination of accuracy and robustness. Robustness refers to the number of times a tracker needs to be initialized.

Table 1: EAO and robustness associated with different components of our proposed tracker on the VOT2017 dataset.

Sl	Ablation	Remark	EAO↑	Robustness↓
· **Template update**				
1	Only SiamFC+	Baseline	0.23	0.49
2	SiamFC+ and UpdateNet	Baseline and Update	0.26	0.40
3	SiamFC+ and Moving Average	Baseline and Linear	0.25	0.44
4	SiamFC+ and Dynamic Update	Ours without Constraint	0.27	0.41
5	SiamFC+ and Dynamic Constr	Ours with INQ. Constraint	0.29	0.38
· **Generative Modelling**				
6	Generated Template Update	5) + Generated Template	0.29	0.37
7	Generated Model and Blend	6) + Tracker Output Blend	0.30	0.37
· **Change Detection**				
8	Change Detection	7) + No Update on Drastic Change	0.31	0.34

Table 2: Accuracy of our proposed and state-of-art trackers on the OTB-50, OTB-100, VOT2016 and VOT2017 datasets.

Tracker	OTB2013		OTB2015		VOT2016			VOT2017		
	AUC↑	Prec↑	AUC↑	Prec↑	EAO↑	A↑	R↓	EAO↑	A↑	R↓
SINT, CVPR-16 Tao et al.	0.64	0.85	-	-	-		-	-		-
SiamFC, ECCV-16 Bertinetto et al. [2016]	0.61	0.81	0.58	0.77	0.24	0.53	0.46	0.19	0.5	0.59
DSiam, ECCV-17 Zhu et al.	0.64	0.81	0.64	0.81	-		-	-		-
StructSiam, ECCV-18 Zhang et al. [c]	0.64	0.88	0.62	0.85	-	-	-	-	-	-
TriSiam, ECCV-18, Dong and Shen	0.62	0.82	0.59	0.78	-	-	-	0.2		-
SiamRPN, CVPR-18 Li et al. [a]	-	-	0.64	0.85	0.34	0.56	0.26	0.24	0.49	0.46
SE-Siam, WACV-21 Sosnovik et al.	0.68	0.90	0.66	0.88	0.36	0.59	0.24	0.27	0.54	0.38
SiamFC+, CVPR-19 Zhang and Peng	0.67	0.88	-	-	0.30	0.54	0.26	0.24	0.49	0.46
SiamRPN++, CVPR-19 Li et al. [b]	-	-	0.69	0.89	0.46	0.64	0.20	0.41	0.60	0.23
Adaptive SiamFC+ (ours)	0.68	0.89	0.67	**0.89**	**0.39**	0.56	0.21	**0.31**	0.52	**0.34**
Adaptive SiamRPN++ (ours)	-	-	**0.71**	0.87	0.47	0.61	**0.19**	**0.44**	0.58	**0.21**

b) Ablation Study: We study the contribution of different components of our proposed method on the VOT2017 dataset. In the first part of Tab 1, "Template update," demonstrates our contribution to model adaptation. The second part, "Generative Model," evaluates the contribution of the generative model in the template update. Finally, the "Change Detection" part shows the effect of change detection on tracking EAO. In order to evaluate the template update part, we compare the results of the baseline Zhang and Peng which is also our backbone. The template update mechanism uses the output from tracker instead of the generative model instead of $\widehat{\varphi}^t$ in the template update network. We implement Zhang et al. [b] based model adaptation for the baseline Zhang and Peng and moving average based linear update as in Zhu et al. is compared with our proposed update method "Dynamic Update" (with attention), which refers to training without the inequality constraint discussed above. Number 5) in the table refers to the experiment where template update is used with inequality constraints. It can be seen that using the inequality constraint alone and our template update mechanism has improved the overall Robustness of the tracker as indicated by the robustness score(lower the score more robust the tracker is). 6) and 7) in the Tab. 1 uses the output from generative model to feed $\widehat{\varphi}^t$. Since the generative model's output is a bit blurry in 7) we blend it with tracker output extracted target template image to obtain a sharper image. Such blending has been shown to improve the result further. We detect drastic changes in the model via the regularity score of the tracker. The change detection will help prevent noisy updates during drift or occlusion; this is shown in 8) where no updates were made during drastic changes.

c) Comparison with State-of-Art: We compare our proposed template update method implemented on SiamFC+ Zhang and Peng back-end against popular Siamese methods bench marked on OTB-50,OTB-100,VOT16,17 datasets. Similar to the benchmarking method in SE-SiamFC Sosnovik et al. we have selected the Siamese trackers for direct comparison with ours. It is important to note that our back-end Siamese tracker, training procedure, sample selection, Etc., are the same as Sosnovik et al.. OTB benchmark uses AUC, which signifies the average overlap over the dataset, and Precision (Prec) signifies the center distance error between object and tracker bounding box. We can see that our method performs competitively with Sosnovik et al. on OTB dataset shown in Tab.2 . It is important to note that OTB does not re-initialize the tracker on failure, and in addition, OTB does not consider track failures into the final evaluation.

On the other hand, the VOT dataset uses Expected average Overlap (EAO), Robustness (R), and Accuracy (A) as metrics. Particularly Robustness is interesting as it indicates some measure on tracker drift in a given dataset. EAO combines tracking accuracy and Robustness, and hence it is a better indicator of tracker performance than just AUC. We can see from the Tab.2 our method outperforms SOA by 4% and outperforms the baseline SiamFC+ Zhang and Peng by 7% on EAO. The results show that our proposed method would enable the tracker to track for longer periods before complete failure compared to the other methods we compare.

To show drastic changes during tracking, we plot the IOU "overlap" (intersection over union for tracking bounding box over ground truth) and the regularity score produced by our change detector. In Fig 4 blue line indicates IOU for our proposed tracker. The thumbnails at the bottom indicate cutouts of the ground truth bounding box around the object being tracked. The video example is from "basketball" of the VOT17 dataset. It can be observed that the regularity score produced by our change detector is low during frames that have partial occlusion and during clutter around the background.
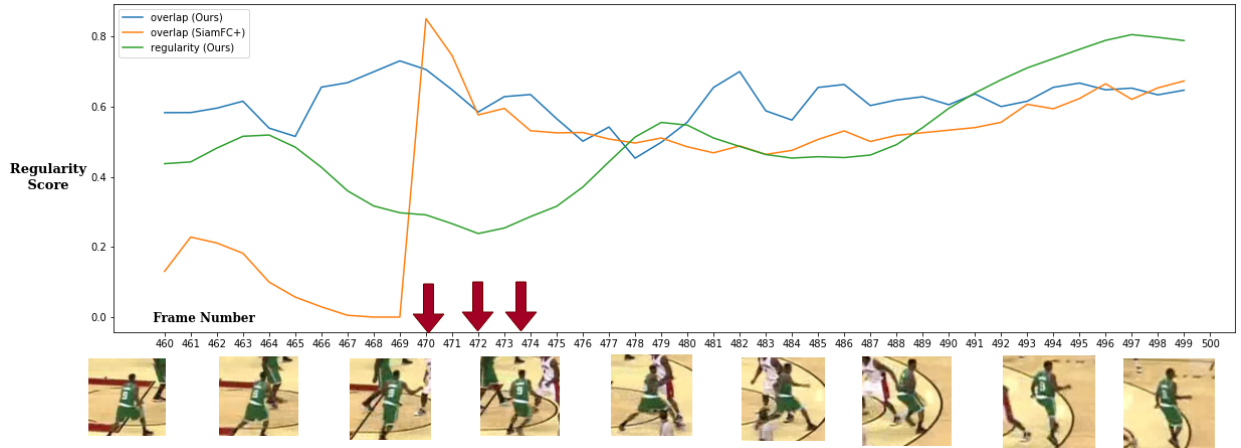

Figure 4: Visualization of tracker accuracy in terms of instantaneous overlap (overlap) of tracker output with ground truth bounding box with video frame number on x axis. We show the results for the trackers with our proposed model update and the baseline SiamFC. Red arrow on the x-axis indicates points of drastic changes.

5 Conclusion

Adaptive Siamese trackers commonly rely on the tracker's output to update the target model. In this paper, we have identified shortcomings with this approach, and proposed a generative model to predict a synthetic target template based on the appearance of several templates from previous time steps. Since the generative model learns the future template from the distribution over past time steps, it suppresses stochastic noise. We also propose a change detection mechanism to avoid noisy updates during abrupt changes in target appearance. Our proposed method can be integrated into any Siamese tracker, and results achieved on VOT16, VOT17, OTB-50, and OTB-100 datasets indicate that it can provide a high level of robustness (can track for a longer period before drifting) compared to state-of-art adaptive and baseline trackers.

References

S. Salti, A. Cavallaro, and L. Di Stefano. Adaptive appearance modeling for video tracking: Survey and evaluation. *IEEE Transactions on Image Processing*, 21(10):4334–4348, 2012.

S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng, S. L. Hicks, and P. H. S. Torr. Struck: Structured output tracking with kernels. *IEEE Trans. PAMI*, 38(10):2096–2109, 2016.

J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

G. Nebehay and R. Pflugfelder. Consensus-based matching and tracking of keypoints for object tracking. In *WACV 2014*, March . doi:10.1109/WACV.2014.6836013.

X. Wang, M. O'Brien, C. Xiang, B. Xu, and H. Najjaran. Real-time visual tracking via robust kernelized correlation filter. In *ICRA 2017*.

Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. *arXiv:1606.09549*, 2016.

Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *CVPR2020*, a.

Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR 2018*, a.

Yuhong Li and Xiaofan Zhang. Siamvgg: Visual tracking using deeper siamese networks. 2019.

Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR 2019*.

Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV 2020*, a.

Zheng Zhu, Qiang Wang, Li Bo, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV 2018*.

Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV 2019*, b.

Tianyu Yang and Antoni B Chan. Learning dynamic memory networks for object tracking. In *ECCV 2018*.

Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR 2013*.

M. Kristan and et al. The visual object tracking vot2017 challenge results. In *ICCVW 2017*.

Matej Kristan and et al. The sixth visual object tracking vot2018 challenge results, 2018.

Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *CVPR 2016*.

Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR2019*.

Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR2019*, a.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6182–6191, 2019.

Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *CVPR 2018*.

Bineng Zhong, Bing Bai, Jun Li, Yulun Zhang, and Yun Fu. Hierarchical tracking by reinforcement learning-based searching and coarse-to-fine verifying. *IEEE Transactions on Image Processing*, 28(5):2331–2341, 2018.

Elvan Duman and Osman Ayhan Erdem. Anomaly detection in videos using optical flow and convolutional autoencoder. *IEEE Access*, 7:183914–183923, 2019.

Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR 2017*.

Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *CVPR 2018*.

Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *ICCV2017*, b.

Yingjie Yao, Xiaohe Wu, Lei Zhang, Shiguang Shan, and Wangmeng Zuo. Joint representation and truncated inference learning for correlation filter based tracking. In *ECCV 2018*.

Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR 2020*, b.

Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR 2016*.

Yao Tang, Lin Zhao, Shanshan Zhang, Chen Gong, Guangyu Li, and Jian Yang. Integrating prediction and reconstruction for anomaly detection. *Pattern Recognition Letters*, 129:123–130, 2020.

Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *ICM2017*.

Arulkumar Subramaniam, Athira Nambiar, and Anurag Mittal. Co-segmentation inspired attention networks for video-based person re-identification. In *ICCV 2019*.

Ivan Sosnovik, Artem Moskalev, and Arnold WM Smeulders. Scale equivariance improves siamese tracking. In *WACV2021*.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2019.

Yunhua Zhang, Lijun Wang, Jinqing Qi, Dong Wang, Mengyang Feng, and Huchuan Lu. Structured siamese network for real-time visual tracking. In *ECCV 2018*, c.

Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *ECCV 2018*.

B Li, W Wu, Q Wang, F Zhang, J Xing, and J SiamRPN+ Yan. Evolution of siamese visual tracking with very deep networks. In *CVPR 2019*, b.