

Leveraging the Power of Data Augmentation for Transformer-based Tracking

Jie Zhao¹, Johan Edstedt², Michael Felsberg^{2,†}, Dong Wang^{1,†}, Huchuan Lu¹
¹Dalian University of Technology ²Linköping University

Abstract

Due to long-distance correlation and powerful pre-trained models, transformer-based methods have initiated a breakthrough in visual object tracking performance. Previous works focus on designing effective architectures suited for tracking, but ignore that data augmentation is equally crucial for training a well-performing model. In this paper, we first explore the impact of general data augmentations on transformer-based trackers via systematic experiments, and reveal the limited effectiveness of these common strategies. Motivated by experimental observations, we then propose two data augmentation methods customized for tracking. First, we optimize existing random cropping via a dynamic search radius mechanism and simulation for boundary samples. Second, we propose a token-level feature mixing augmentation strategy, which enables the model against challenges like background interference. Extensive experiments on two transformer-based trackers and six benchmarks demonstrate the effectiveness and data efficiency of our methods, especially under challenging settings, like one-shot tracking and small image resolutions. Code is available at <https://github.com/zj5559/DATr>.

1. Introduction

With the development of deep models, many visual object tracking (VOT) works [7, 10, 23, 36, 47] focus on designing effective tracking frameworks with modern backbones. Some large-scale tracking datasets with high-quality manual annotations [14, 19, 31] are also developed to satisfy these data-driven models. However, a crucial issue is long neglected, that is, appropriate data augmentation is the cheapest strategy to further boost the tracking performance. We notice that most trackers follow similar data augmentation strategies, which are combinations of random cropping and several image transformations, like flip and blur. State-of-the-art (SOTA) transformer-based methods also apply the same pattern as prior works based on convolutional neural networks (CNN). Bhat *et al.* [3] demonstrated that these general data augmentations (GDA) play

[†] Corresponding authors.

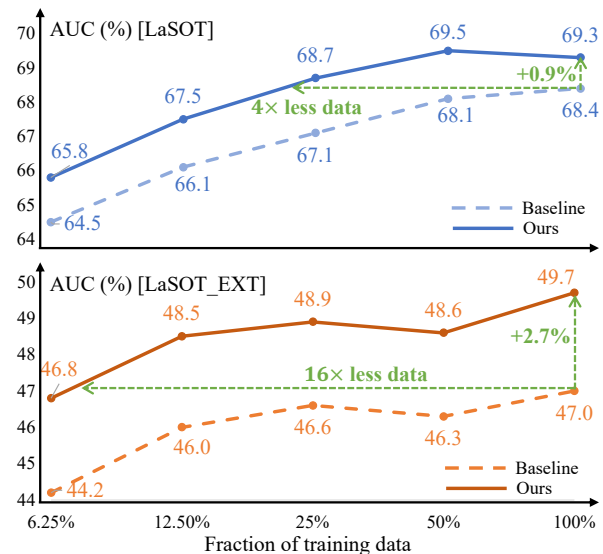


Figure 1. **Data-efficiency comparison under different volumes of training data.** Results are averaged over 3 random seeds.

an important role on CNN-based trackers. However, considering the substantial difference between CNN and transformer models, and powerful capabilities of transformer models themselves, what is the impact of GDAs on SOTA transformer-based trackers? We think this is a question worth exploring. While it has been demonstrated in several works [8, 16, 26, 44, 46] that well-designed data augmentation is useful for multiple computer vision tasks, few works apply the latest data augmentations or customize suitable approaches for VOT.

In this paper, we perform comprehensive experiments to explore the impact of GDAs on transformer-based trackers, including the pure transformer tracker and the hybrid CNN-Transformer tracker. Different from the conclusion in [3], our experiments imply that most common GDAs have limited effectiveness for these SOTA trackers. We also gain the insight that while models can benefit from increasing jitter for random cropping, large jitters will degrade performance. Moreover, as shown in Fig. 2, we find that in addition to the sequence’s own challenges, previous inaccurate predictions also cause difficult search patches with huge scale variations (Left) and boundary targets (Middle). Background interference is also challenging for SOTA trackers (Right).

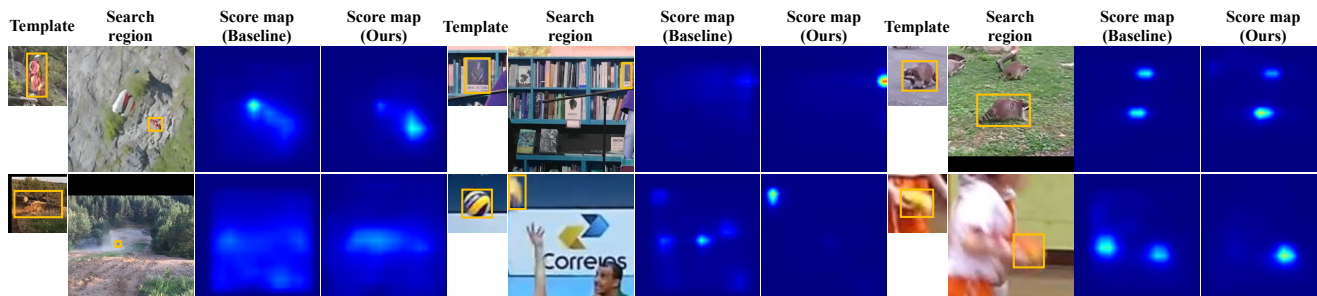


Figure 2. **Qualitative comparison of score maps on hard cases.** Left: Huge scale variations. Middle: Boundary samples caused by previous inaccurate prediction or fast motion. Right: Interference from the background. Better viewed with zoom-in.

Motivated by these observations, we propose two data augmentation approaches customized for VOT. First, we propose an optimized random cropping (ORC) consisting of a dynamic selection mechanism of search radius factor γ and simulation of boundary samples. The former enriches samples from the perspective of context via two-step randomness, which enables the model more robust to scale variations (Fig. 2 (left)), and furthermore makes the model flexible to different γ during inference. The latter helps the model recover fast from failure cases and deal with challenges like fast motion better (Fig. 2 (middle)). Second, we propose a token-level feature mixing augmentation (TFMix). Token features of another object are mixed into the original search features as a distractor. This method makes the model better able to cope with complex background interference (Fig. 2 (right)).

Experiments in Sec. 5 demonstrate that our methods not only further boost modern trackers’ performance, especially under challenging settings, but also unbind strong association for specific value of γ between training and inference. Furthermore, to explore the data efficiency benefit from our methods, we use different volumes of data for model training, *i.e.* randomly choosing a fraction of sequences from each training dataset. Since we find that reducing the numbers of training sample pairs for settings with small data volumes has little effect on the performance, we follow the same number of sample pairs as the baseline setting (OSTrack256). As shown in Fig. 1, using only 6.25% of the data, our methods achieve comparable result on LaSOT_EXT to the baseline trained with full data.

The main contributions of this work are as follows:

- We perform systematic experiments to explore the impact of General Data Augmentations (GDA) on transformer trackers, including the pure transformer tracker and the hybrid CNN-Transformer tracker. Results show GDAs have limited effects on SOTA trackers.
- We propose two Data Augmentation methods based on challenges faced by Transformer-based trackers, DATr for short. They improve trackers from perspectives of adaptability to different scales, flexibility to

boundary targets, and robustness to interference.

- We apply DATr to two transformer trackers on six tracking benchmarks, demonstrating the effectiveness and generalization of DATr, especially for sequences with challenges and unseen classes. Experiments on CNN backbones further show the significant generalization effect of our optimized random cropping.

2. Related Work

2.1. Visual Object Tracking

In terms of the types of backbones, tracking methods have gone through three stages of evolution, *i.e.* traditional approaches [5, 18, 20] using hand-crafted features, CNN-based methods [1, 2, 23, 47], and transformer-based methods [7, 9, 27, 41, 43]. Among them, SiamRPN++ [23] and SiamDW [47] analyzed the negative effect of large receptive field and padding issue caused by the deep CNN, and investigated proper architectures to make the tracking benefit from very deep CNN. To make up for the locality of CNN, Chen [7] *et al.* developed a transformer-based correlation module, which can establish long-distance associations between the template and search region. Recently, several works, *e.g.* OTrack [43] and MixFormer [9], combined the feature extraction and fusion modules into a whole through a pure transformer architecture, boosting the tracking performance to a new level.

2.2. Data Augmentation in Tracking

Most previous works in tracking focus on designing effective model architectures, or integrating modern backbones into tracking framework. In contrast, despite data augmentation playing a crucial role in the performance of trackers, far less attention has been paid to this topic.

Augmentations for CNN-based trackers. Zhu *et al.* [50] investigated the important role of diverse training samples. Bhat *et al.* [3] compared performance gains from GDAs for shallow and deep features and found that deep CNNs particularly benefit from augmentation. Motivated by model robustness to rapid motion, several works [3, 38, 50] em-

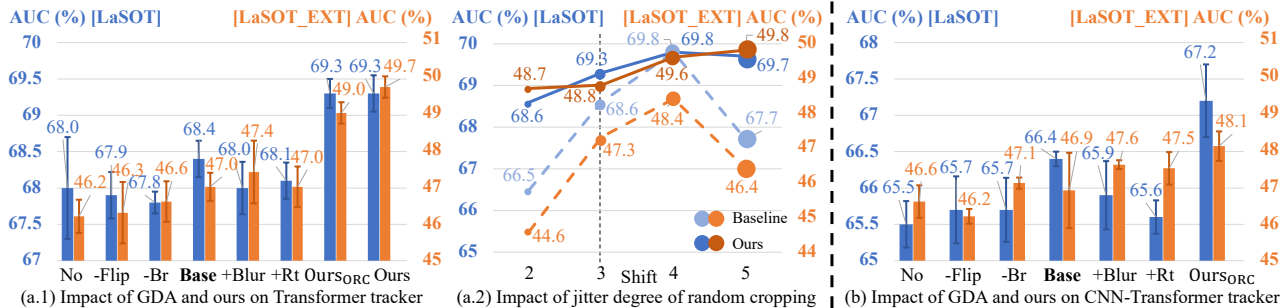


Figure 3. **Systematic analysis of GDAs, and comparison with ours on LaSOT (blue) and LaSOT_EXT (orange).** (a.1) and (b) compare the impact of GDA and ours on the pure transformer tracker and the CNN-Transformer tracker, respectively. Results imply a limited effectiveness of GDAs for these SOTA trackers, while ours improve their performance significantly on each benchmark. (a.2) shows the existing random cropping causes model degradation under small and large jitter, while ours are stable for different jitter settings.

phasized the impact of blur augmentation.

Augmentations for transformer-based trackers. Transformers have been found to exhibit different properties [33] and be more robust to perturbations [4, 34] than CNNs. Despite this, previous transformer-based trackers [9, 28, 43] still use similar data augmentation approaches as for CNNs, and the impact of these augmentations has not been investigated. In contrast to previous works, we systematically investigate the role of GDAs for modern transformer-based trackers (see Sec. 3). Motivated by experimental observations, we further propose two data augmentation approaches based on challenges faced by modern transformer-based trackers (see Sec. 4).

2.3. Image Mixing Augmentation

In the context of computer vision, in addition to the basic geometric transformations (*e.g.* rotation, scaling, shear, flip), and photometric transformations (*e.g.* saturation, grayscale, color-jittering), a multitude of augmentations obtain diverse data via mixing different images. For example, MixUp [46] blends images pixel by pixel. CutMix [44] replaces contents of a random area with a patch cropped from another image. Customized for transformer models, TokenMix [26] mixes images in units of tokens. The effectiveness of these mixing methods has been demonstrated in many tasks, such as object detection [12, 37], instance segmentation [16], and video classification [45], but few works integrate this type of methods into VOT. To the best of our knowledge, the only work to apply a similar strategy to tracking is [24], which performs crop-transform-paste operations on images for self-supervised tracking. Unlike this, we propose a token-level feature mixing strategy to simulate background interference.

3. Analysis of General Data Augmentation

General data augmentations (GDA) are ubiquitously used in tracking. As shown in Tab. 1, we summarize data augmentation strategies of 40 trackers published in the past five years, and find that most trackers apply random

Table 1. **Usage count of each data augmentation in trackers published in five years.** (“RC” indicates random cropping.)

Models	Grayscale	RC	Flip	Bright	Blur	Rotate
CNN (28)	10	28	4	23	13	1
Transformer (12)	11	12	8	11	0	0

cropping along with several similar combinations of image transformations. Especially for recent transformer-based trackers, all of them follow a similar augmentation pattern as prior CNN-based works.

Although Bhat *et al.* [3] has shown the importance of these GDAs on deep CNN models, the efficacy of GDAs has as of yet not been investigated for modern transformer trackers. Hence, we pose the following question: **What is the impact of GDAs on transformer-based trackers?** To explore the answer, we perform systematic experiments described in Sec. 3.1, and analyze results in Sec. 3.2.

3.1. Experimental Settings

As shown in Fig. 3, we choose OTrack [43] (see (a.1) and (a.2)) and STARK [41] (see (b)) as baselines to represent the pure transformer tracker and the hybrid CNN-Transformer tracker, respectively, and evaluate all models on LaSOT [14] and LaSOT_EXT [13]. Their official augmentations are grayscale, random cropping, brightness jitter, and horizontal flip, which are also most used for other transformer-based trackers (see Tab. 1). Since grayscale is required to make the model robust to grayscale sequences, while random cropping prevents models from overfitting, we consider the two approaches as necessary. The model trained with only these two approaches is represented as “No”, while the official setting is denoted as “Base”.

To explore the impact of the other methods, along with blur and rotation which are applied by some CNN-based trackers, we remove (horizontal flip or brightness jitter) or add (blur or rotation) each augmentation on the basis of “Base”, represented as “-Flip”, “-Br”, “+Blur”, and “+Rt” in Fig. 3 (a.1) and (b), respectively. Considering stability, we run each experiment three times with different random

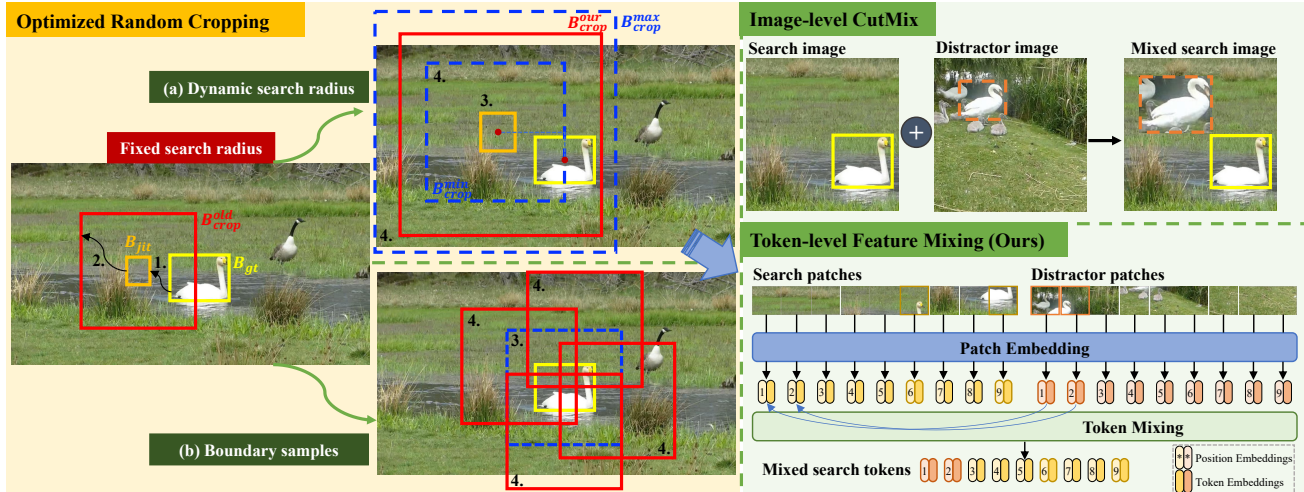


Figure 4. **Illustration of our customized data augmentation strategies.** Left: Optimized random cropping including (a) dynamic selection mechanism for the search radius factor γ , and (b) simulation of boundary samples. It does not only enrich the diversity of samples from two-step randomness, but renders also the model insensitive to the parameters. Numbers from 1. to 4. indicate the order of cropping steps. Right: Classical image-level CutMix (top), and our token-level feature mixing augmentation (bottom).

seeds, and record their average performance with the standard deviation (STD), which is illustrated as error bars. Besides, to avoid negative effects from inappropriately tuned probability and magnitude, we test different settings for blur and rotation, and use the best-performing parameters in our systematic experiments.

In addition, we also investigate the impact of the jitter degree of random cropping. In Fig. 3 (a.2), we set different jitter degree of random cropping by adjusting the magnitude of scale and shift. The size of circles represents the value of scale¹, and the black dotted line indicates the official setting of the baseline (OSTrack).

3.2. Observations and Analysis

Different types of GDA. Experiments in Fig. 3 (a.1) and (b) imply that these GDAs seem to have limited effectiveness for the SOTA transformer-based tracker. Taking results on different benchmarks and error bars into account, we can conclude that these GDAs do not provide substantial improvement but only slight fluctuations up and down the baseline models.

Different jitter degree of random cropping. From the trend of dotted curves in Fig. 3 (a.2), we find that a proper setting of random cropping can significantly improve the tracking model. The model can benefit more from larger jitter, e.g. Shift₄ vs. Shift₂. However, further increasing the jitter degree will cause model degradation, e.g. Shift₅.

Analysis. Due to global correlation and models [17] pre-trained on large-scale datasets, transformer models trained without GDA (see “No” in Fig. 3 (a.1) and (b)) can already address most situations which are difficult for CNNs. However, we observe that challenges like background interfer-

ence are still difficult for modern trackers (see Fig. 2), and cannot be simulated by aforementioned GDAs. Therefore, customized augmentations based on unsolved challenges are needed to further improve SOTA transformer trackers.

As for random cropping, we can conclude from the dotted curves in Fig. 3 (a.2) that various samples with different target positions and scales are conducive to training models. However, in the existing random cropping strategy with fixed search radius factor γ_{fix} , the shift parameter should not be set larger than γ_{fix} to avoid uninformative samples, i.e. the object is outside the patch. Otherwise, these uninformative samples would pollute the training set and cause model degradation, e.g. results of Shift₅ where $\gamma_{fix} = 4$. Therefore, the existing random cropping strategy with a fixed context scope, does not only limit the diversity of samples, but also cause the parameter sensitivity.

4. Data Augmentation Customized for VOT

Motivated by the analysis in Sec. 3.2, we propose two customized data augmentation approaches. First, optimized random cropping (ORC) is proposed, including dynamic selection for search radius factor, and simulation of boundary samples, where the partial region of the object stays at the boundary of the search patch. Second, we propose a token-level feature mixing strategy (TFMix) to simulate unsolved challenges, like background interference. We describe these two augmentations in Sec. 4.1 and Sec. 4.2, respectively.

4.1. Optimized Random Cropping

Existing trackers essentially treat tracking as a local matching problem between templates and search regions. The local search region is decided by the predicted target location of the previous frame, and a fixed search radius

¹The scale value is traversed from 0.15 to 0.45

factor γ . To maintain the distribution consistency of samples, random cropping with the same value of γ as inference is applied in the training phase. Consider for instance the cropping strategy in the transformer-based methods as an example², as shown in Fig. 4 (left), the existing random cropping strategy has two steps, *i.e.*, jitter the groundtruth B_{gt} via random shifting and scaling (“1.” in Fig. 4), and crop the search region B_{crop}^{old} based on the jittered bounding box B_{jit} as well as a fixed γ (“2.” in Fig. 4).

There are several disadvantages to this strategy. First, only one random step (“1.” in Fig. 4) is performed to support diversity of samples. Second, the degree of shift is constraint by γ_{fix} to avoid uninformative samples, which leads the training process to be sensitive to parameters. In addition, training with a fixed γ makes the model inflexible, *i.e.* forcing the model to be specific to the same γ in inference, shown as Tab. 4 (discussed in Sec. 5.2).

In this paper, we propose an optimized random cropping strategy to address these issues. As shown in Fig. 4 (a), to enrich the diversity of training samples, and also unbind the model from the strong association with specific γ in inference, we first turn the fixed γ during training into a dynamic selected value from γ_{min} to γ_{max} . The maximum and minimum values are used to limit the proportion of context in search regions. Otherwise, the target will be very small or large in the resized search patch. Furthermore, to avoid uninformative samples, we calculate the practical minimum search radius factor γ_{min}^p based on the distance between center locations of B_{gt} and B_{jit} . If γ_{min}^p is larger than γ_{max} , we consider the current B_{jit} to be invalid, and retry to find a proper one. Through this simple strategy, uninformative samples can be avoided without sacrificing the diversity of the training set. It is worth noting that although the original random cropping strategy can achieve context variation implicitly by B_{jit} , compared with this one-step randomness, our method consists of two random steps, *i.e.* B_{jit} and dynamic γ , which are able to obtain samples with more diverse scales and contexts.

Besides, considering that objects often appear at the boundary or even partially outside search regions in some failure cases and challenges like fast motion. we simulate such boundary samples with probability P_b , shown as Fig. 4 (b). We first calculate the search region (blue dashed box) based on B_{gt} , and then shift it to a random direction until the target is partially at the boundary. It helps models cope with boundary targets more accurately.

The procedure of our ORC is described as Algorithm 1. I_s denotes the processed search frame, D_{jit} and S_{jit} represent the magnitude of random shifting and scaling, P_b identifies the probability of boundary samples, ct_s and ct_{jit} represent center locations of B_{gt} and B_{jit} . Due to dynamic γ

²Prior Siamese-based trackers [1, 23, 48] apply similar parameter as γ to fix the context scope.

Algorithm 1 Optimized Random Cropping

Input: $I_s, B_{gt}, \gamma_{min}, \gamma_{max}, D_{jit}, S_{jit}, P_b$

Output: B_{crop}, γ

```

1:  $\gamma = \text{Random}(\gamma_{min}, \gamma_{max});$ 
2: if  $\text{Random}(0,1) < P_b$  then
3:    $B_{crop} = \text{CenterCrop}(B_{gt}, \gamma);$   $\triangleright$  Fig. 4 (b) “3.”
4:    $\text{direction} = \text{Random}(\text{top}, \text{bottom}, \text{left}, \text{right});$ 
5:    $B_{crop} = \text{Move}(B_{crop}, \text{direction});$   $\triangleright$  Fig. 4 (b) “4.”
6: else
7:   while True do
8:      $B_{jit} = \text{Jitter}(B_{gt}, D_{jit}, S_{jit});$   $\triangleright$  Fig. 4 (a) “3.”
9:      $\gamma_{min}^p = \text{MAX}\left\{\frac{2|ct_s - ct_{jit}|_{max}}{\sqrt{w_{jit} \times h_{jit}}}, \gamma_{min}\right\};$ 
10:    if  $\gamma_{min}^p \leq \gamma_{max}$  then
11:       $\gamma = \text{Random}(\gamma_{min}^p, \gamma_{max});$   $\triangleright$  Fig. 4 (a) “4.”
12:       $B_{crop} = \text{CenterCrop}(B_{jit}, \gamma);$ 
13:      Break;
14:    end if
15:  end while
16: end if

```

and boundary samples, our ORC can enrich the diversity of samples from different perspectives, such as the context scope, target positions and scales, while avoiding uninformative samples. Experiments in Sec. 5 demonstrate ORC improvements to performance and γ flexibility.

4.2. Token-level Feature Mixing

Background interference is one of the main challenges for modern trackers, but such samples are not the focus of GDAs, which might be a potential reason for their limited effectiveness. Recent augmentations like CutMix [44] can be an option to synthesize hard samples with background interference, as shown in Fig. 4 (top-right). However, such image mixing tends to trap the model in overfitting to sharp border effect. To mitigate this issue and consider the token mechanism of transformer models, we propose a token-level feature mixing method as shown in Fig. 4 (bottom-right). A search patch with the object \mathcal{O}_s , and a distractor patch with another object \mathcal{O}_d are first cropped and processed by a linear projection, we then transfer distractor tokens $T_d^{\mathcal{O}_d}$ belonging to \mathcal{O}_d and replace search tokens $T_s^{\mathcal{O}_d}$ in corresponding positions, represented as

$$T_s^{\mathcal{O}_d} = \frac{T_d^{\mathcal{O}_d} - \text{mean}_{\mathcal{O}_d}}{\text{std}_{\mathcal{O}_d}} \text{std}_{\mathcal{O}_s} + \text{mean}_{\mathcal{O}_s}. \quad (1)$$

Distractor tokens $T_d^{\mathcal{O}_d}$ will be normalized before transferring to alleviate huge discrepancy between \mathcal{O}_s and \mathcal{O}_d , where $\text{mean}_{\mathcal{O}_d}$ and $\text{std}_{\mathcal{O}_d}$ represent the global mean and standard deviation of the object tokens. To increase the difficulty of samples, we preferentially select \mathcal{O}_d from the same category as \mathcal{O}_s . Besides, an occluded threshold is

used to control the occluded degree of \mathcal{O}_s .

5. Experiments

To investigate the effectiveness of our augmentations DATr, we apply them to two SOTA transformer trackers, MixFormer [9] and OSTRack [43]. Besides, we also apply our ORC to a hybrid CNN-Transformer tracker STARK [41], and a CNN-based tracker SiamFC++ [40] to demonstrate its generalization ability to CNN backbones.

5.1. Implementation Details

We implement our data augmentations in Python with PyTorch. All experiments are trained using four NVIDIA A100 GPUs. For our data augmentations, we set the probability of boundary samples, P_b , to 0.05. To keep the dynamic selection range of the search radius factor γ symmetrical to the fixed value in inference, we set it as [2,6] when $\gamma = 4$ in inference, and [4,6] when $\gamma = 5$. As for the mixing, our TFMix augmentation is triggered every 11 epoches, and the occluded threshold is set to 0.5. It is worth noting that our DATr can augment both video and image datasets. For a fair comparison, we adopt the same training settings as for the baseline to retrain each tracking model with and without our augmentations, where training datasets include LaSOT, GOT-10k [19], TrackingNet [31], and COCO [25].

5.2. Ablation Study and Analysis

Using OSTRack with 256 image resolution as the baseline tracker, we perform a series of ablation study on LaSOT and LaSOT_EXT to demonstrate the effectiveness of our approaches from different aspects. LaSOT contains 280 sequences with 70 categories, which are the same as its training subset. In contrast, LaSOT_EXT is composed of 150 very challenging sequences with 15 unseen categories. One-pass evaluation is performed on both benchmarks with three metrics: the success rate (AUC), precision (Pre), and normalized precision (P_{norm}). AUC represents the ratio of successfully tracked frames, while Pre and P_{norm} represent the distance of center points between the groundtruth and the predicted result. P_{norm} is normalized with the target scale, which is stable to target size and image resolution.

Impact of each component. As shown in Tab. 2, our optimized random cropping can obtain 2.0% and 0.9% AUC gains (Base vs. ②) on LaSOT_EXT and LaSOT, respectively. Among them, dynamic γ mechanism increases AUC with 1.7% on LaSOT_EXT, and simulating boundary samples can further improve AUC to 69.3% on LaSOT. Due to generating challenging samples, our TFMix boosts the performance to 49.7% on LaSOT_EXT.

Different mixing strategies. To demonstrate the effectiveness of the proposed TFMix, we compare it with dif-

Table 2. **Impact of each proposed component on AUC³.**

Method	Dynamic γ	Boundary	TFMix	LaSOT	LaSOT_EXT
Base				68.4	47.0
①	✓			68.9	48.7
②	✓	✓		69.3	49.0
③	✓	✓	✓	69.3	49.7

Table 3. **Comparison of different mixing strategies³**, including different image-level mixing methods, and late feature mixing.

Mixing Strategy	LaSOT			LaSOT_EXT			
	AUC	P_{norm}	Pre	AUC	P_{norm}	Pre	
Image	Bbox	69.0	79.0	75.2	48.6	59.3	55.5
	Mask	69.4	79.7	75.8	48.3	58.9	55.1
	Token	69.0	79.1	75.2	49.0	59.7	55.7
Feature	Late	68.8	79.0	75.1	49.0	59.7	55.4
	Early (Ours)	69.3	79.3	75.3	49.7	60.4	56.6

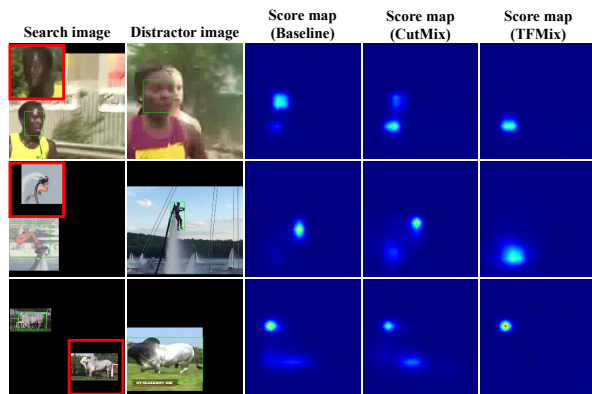


Figure 5. **Comparison of discriminative ability between image-level CutMix and TFMix.** Templates are framed by red boxes.

ferent mixing strategies, including several different image-level mixing methods, and late feature-level mixing. The same parameter settings are used for a fair comparison. For the image mixing with bbox, we simply use bounding box annotations to crop a rectangle area of the distractor, while for the image mixing with mask, we first obtain mask annotations of all training datasets from Alpha-Refine [42], and paste the distractor itself without any context. The token image mixing is similar to TokenMix [26], we set the same token size as for the model, and randomly mix 30% to 50% of them between search patches and distractor patches. As for the late feature mixing, our TFMix can be considered as an early-stage feature mixing, since the mixing is performed before the feature extraction and fusion. In contrast, the late mixing delays this operation until after the feature fusing. In the feature extraction and fusion stage, token interactions not only happen between the template and search patch, but also occur between context tokens in the search patch itself. Therefore, late feature mixing will miss the core interaction between the original object and the extra distractor. As shown in Tab. 3, our TFMix is superior to other mixing strategies, especially on the most challenging

³Reported results are averaged over 3 random seeds.

Table 4. **Adaptability comparison to different γ in inference.**

$\gamma_{\text{train}} \rightarrow \gamma_{\text{test}}$	LaSOT				LaSOT_EXT			
	AUC	P _{norm}	Pre	STD	AUC	P _{norm}	Pre	STD
4 \rightarrow 3	58.9	67.5	64.9		40.1	49.4	46.1	
4 \rightarrow 4	68.6	78.1	74.4	5.02	47.3	57.4	53.1	5.12
4 \rightarrow 5	61.5	68.7	63.8		37.4	44.2	38.7	
3 \rightarrow 3	67.5	76.3	72.9		43.6	52.2	47.8	
4 \rightarrow 4	68.6	78.1	74.4	0.64	47.3	57.4	53.1	2.40
5 \rightarrow 5	67.5	77.4	72.7		48.1	58.5	54.4	
Dyn. \rightarrow 3	67.1	76.7	72.6		44.2	54.0	50.0	
Dyn. \rightarrow 4	69.3	79.4	75.5	1.11	48.8	59.4	55.6	2.52
Dyn. \rightarrow 5	68.4	78.3	73.7		48.3	58.6	54.4	

benchmark, LaSOT_EXT.

Moreover, we compare the discriminative ability gained from image mixing with bbox (CutMix), and our TFMix. As shown in Fig. 5, when distractor tokens are mixed into search patches via Eq. 1, the baseline tracker is prone to being confused by distractors. CutMix improves this phenomenon to some extent (see first row), while the last column shows that our TFMix promotes the model be more discriminative to distractors.

Adaptability to different γ in inference. Different from prior training, since the proposed dynamic γ mechanism enriches training samples from the perspective of contextual information, the model should be more adaptive to search patches cropped with different γ in inference. To investigate the validity of this conjecture, we conduct three sets of experiments shown as Tab. 4, where “ $\gamma_{\text{train}} \rightarrow \gamma_{\text{test}}$ ” represents that the search radius factor is set to γ_{train} in the training phase, and γ_{test} in the inference. “Dyn.” represents to train the model using our dynamic γ mechanism.

We can see that the model trained with a fixed γ_{train} performs extremely poorly when faced with different γ_{test} in the inference (see results of “4 \rightarrow i ”). The AUC standard deviation (STD) of the first set is higher than 5 on both two benchmarks. While in the second set (“ $i \rightarrow i$ ”), well performance with lower STD under different γ_{test} can be obtained when we keep the consistency of γ in the training and inference. This phenomenon shows that the original cropping strategy using fixed γ_{train} establishes a strong association of γ between the training and inference, which hinders the adaptability of models to scale variations, especially caused by previous inaccurate prediction (see Fig. 2 (left)).

In contrast, our model effectively unbinds this kind of association due to the proposed dynamic γ in the training phase. Our model (“Dyn. $\rightarrow i$ ”) performs well on all different γ_{test} , and has a comparable low STD with the second set. We think this characteristic of our approach not only helps the model to be more robust to scale variations, but also provides a new insight for future works related to dynamic search in the inference, like [49].

Stability to different magnitudes of jitter. As concluded in Sec. 3.2, tracking models cannot perform well under small and very large jitter settings in the training phase.

Table 5. **Generalization of our methods to CNN backbones.**

AUC	STARK +ORC +TFMix			SiamFC++ +ORC	
LaSOT	66.4	67.7	66.2	60.4	61.1
LaSOT_EXT	46.5	48.1	46.9	37.7	38.9

Table 6. **Performance comparison on VOT2022 benchmark.**

	EAO	A	R
MixFormer-22k	0.538	0.776	0.838
+DATr (Ours)	0.531 <small>0.7% \downarrow</small>	0.743	0.840
OSTrack256	0.497	0.783	0.788
+DATr (Ours)	0.525 <small>2.8% \uparrow</small>	0.771	0.820
OSTrack384	0.522	0.788	0.799
+DATr (Ours)	0.525 <small>0.3% \uparrow</small>	0.777	0.807
Average gain	+0.8%	-1.9%	+1.4%

To demonstrate that our ORC is more stable to different jitter degrees, we train our model under different jitter settings, as shown in Fig. 3 (a.2). Compared with the original cropping method (light dashed lines), our ORC (dark solid lines) enables the tracking model adapt to varying degrees of jitter. In addition to dynamic γ mechanism, which enriches samples’ diversity, simulating boundary cases can feed models such samples under a small jitter setting. Besides, there is also a check and filter step for uninformative samples in our ORC. Therefore, we can still obtain well-performed model stably even under very small (*e.g.* Shift₂) or very large (*e.g.* Shift₅) jitter. We think this characteristic brings convenience for future works, which prevents models from being too sensitive to jitter parameters.

Generalization capability of our methods. As shown in Tab. 5, in addition to the pure Transformer trackers, our ORC also boosts hybrid CNN-Transformer trackers (*e.g.* STARK) and CNN-based trackers (*e.g.* SiamFC++). However, since our TFMix relies on characteristics of transformer models, *i.e.* global correlation between independent tokens, it shows to be less effective for CNN backbones, causing an average 1.4% AUC decline for STARK. The potential reason might be the strong inductive bias in CNN networks.

5.3. State-of-the-art comparison

We apply our augmentations on two SOTA transformer trackers, MixFormer and OSTrack, and evaluate them on six tracking benchmarks. For the OSTrack, we evaluate its two variants with different image resolutions, represented as OSTrack256 and OSTrack384, respectively.

VOT2022 (STB) [22]. This challenge contains 50 challenging short-term sequences with multiple initial anchor points. The primary measure is the expected average overlap (EAO), which is a principled combination of tracking accuracy (A) and robustness (R). As shown in Tab. 6, our DATr improves three baseline models by 0.8% on average in terms of EAO, especially for OSTrack256, boosting by 2.8% EAO. We can see that our DATr mainly improves models from the perspective of tracking robustness.

Table 7. **State-of-the-art comparisons on five tracking benchmarks.** The top two results are highlighted with **red** and **blue**, respectively.

Method	LaSOT			LaSOT_EXT			GOT-10k			UAV123		NFS	
	AUC	P _{norm}	Pre	AUC	P _{norm}	Pre	AO	SR _{0.5}	SR _{0.75}	AUC	Pre	AUC	Pre
ECO [11]	32.4	33.8	30.1	22.0	25.2	24.0	31.6	30.9	11.1	53.5	76.9	52.2	63.4
SiamFC [1]	33.6	42.0	33.9	23.0	31.1	26.9	34.8	35.3	9.8	46.8	69.4	37.7	44.5
MDNet [32]	39.7	46.0	37.3	27.9	34.9	31.8	29.9	30.3	9.9	52.8	-	42.2	-
SiamRPN++ [23]	49.6	56.9	49.1	34.0	41.6	39.6	51.7	61.6	32.5	59.3	78.2	57.1	69.3
Ocean [48]	56.0	65.1	56.6	-	-	-	61.1	72.1	47.3	57.4	77.8	49.4	61.2
DiMP [2]	56.9	65.0	56.7	39.2	47.6	45.1	61.1	71.7	49.2	64.3	85.1	61.8	73.8
TrDiMP [36]	63.9	-	61.4	-	-	-	67.1	77.7	58.3	66.4	86.9	66.2	79.1
SiamRCNN [35]	64.8	72.2	-	-	-	-	64.9	-	-	64.9	83.4	63.9	-
TransT [7]	64.9	73.8	69.0	-	-	-	67.1	76.8	60.9	68.1	87.6	65.3	78.8
SBT-L [39]	66.7	-	71.1	-	-	-	70.4	80.8	64.7	-	-	-	-
KeepTrack [29]	67.1	77.2	70.2	48.2	-	-	-	-	-	69.7	-	66.4	-
ToMP-101 [28]	68.5	79.2	73.5	45.9	-	-	-	-	-	66.9	-	66.7	-
AiATrack [15]	69.0	79.4	73.8	46.8	54.4	54.2	69.6	80.0	63.2	70.6	-	67.9	-
Sim-L [6]	70.5	79.7	-	-	-	-	69.8	78.8	66.0	71.2	91.6	-	-
MixFormer-22k [9]	68.9	78.5	74.3	49.1	59.6	55.3	70.3	80.0	66.2	69.7	91.0	65.0	79.1
+DATr (Ours)	68.8 _{0.1↓}	78.9 _{0.4↑}	74.6	51.0 _{1.9↑}	61.8 _{2.2↑}	57.3	71.4 _{1.1↑}	81.0 _{1.0↑}	67.6	69.6 _{0.1↓}	90.9 _{0.1↓}	65.8 _{0.8↑}	79.7 _{0.6↑}
OTrack256 [43]	68.6	78.1	74.4	47.3	57.4	53.1	71.4	81.4	67.5	68.2	88.6	65.4	79.6
+DATr (Ours)	69.1 _{0.5↑}	79.1 _{1.0↑}	75.2	49.9 _{2.6↑}	60.6 _{3.2↑}	57.0	72.5 _{1.1↑}	82.3 _{0.9↑}	69.2	70.8 _{2.6↑}	92.4 _{3.8↑}	66.0 _{0.6↑}	81.1 _{1.5↑}
OTrack384 [43]	70.7	80.4	77.0	50.5	61.2	57.4	73.5	83.0	70.6	69.7	90.6	66.3	80.8
+DATr (Ours)	71.0 _{0.3↑}	80.7 _{0.3↑}	77.5	51.8 _{1.3↑}	62.7 _{1.5↑}	59.0	74.2 _{0.7↑}	84.1 _{1.1↑}	71.1	69.7 _{0.0↑}	90.7 _{0.1↑}	65.5 _{0.8↓}	79.9 _{0.9↓}
Average gain	+0.2%	+0.6%	+0.5%	+1.9%	+2.3%	+2.6%	+1.0%	+1.0%	+1.2%	+0.8%	+1.3%	+0.2%	+0.4%

LaSOT and LaSOT_EXT. Compared with LaSOT, its extended dataset LaSOT_EXT is more challenging, and all its categories are unseen from the training set. As shown in Tab. 7, the superiority of our augmentations can be fully reflected on the very challenging benchmark LaSOT_EXT. All of three baseline trackers are improved by 1.9% AUC and 2.3% P_{norm} on average. Our augmentations also bring an average of 0.6 P_{norm} gain on LaSOT.

GOT-10k. GOT-10k is composed of 180 test sequences of which classes are zero-overlapped with its training set. We follow the official one-shot protocol to train all models, where only its training subset is allowed to be used for training. Performance is evaluated by three metrics: average overlap (AO), and success rates with two different thresholds (SR_{0.5} and SR_{0.75}). As shown in Tab. 7, all of our models achieve significant promotion, surpassing baseline trackers by 1.0% AO and 1.2% SR_{0.75} on average.

UAV123 [30] and NFS [21]. These two benchmarks contain 123 sequences captured from the aerial perspective, and 100 sequences, respectively. Results in Tab. 7 show that our DATr obtains 1.3% improvement in terms of precision on UAV123, and also minor increase on NFS.

Discussion and Limitation. In terms of the above experiments, the superiority of our DATr is most evident under challenging settings, like dealing with unseen classes (GOT-10k) or very challenging sequences (LaSOT_EXT), and handling images with small resolution (OTrack256). Since our augmentations are motivated by unsolved challenges and failure cases, our DATr tends to improve models in terms of tracking robustness, instead of accuracy, *i.e.* we aim to locate the target successfully under challenging sit-

uations. This is the potential reason for the slight accuracy decline in Tab. 6, and minor performance gains on some benchmarks, like LaSOT and NFS.

6. Conclusion

In this paper, we systematically analyze the impact of GDAs on modern transformer trackers and propose two customized data augmentations for VOT. First, to improve the adaptability of models to scale variations and boundary targets, we design an optimized random cropping, containing dynamic selection for search radius factor, and simulation of boundary samples. Second, we synthesize hard samples with background interference by a token-level feature mixing strategy. Extensive experiments on two SOTA transformer-based trackers and six benchmarks demonstrate our augmentations enable the model benefit from more diverse and challenging samples, and be more flexible to changes of search radius in inference.

Acknowledgements. The computational resources were provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at C3SE partially funded by the Swedish Research Council grant 2022-06725. This work was supported in part by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP), funded by Knut and Alice Wallenberg Foundation, in part by the strategic research environment ELLIIT funded by the Swedish government, and the Swedish Research Council grant 2022-04266, in part by National Natural Science Foundation of China (Nos.62176041, 62022021), and in part by Fundamental Research Funds for the Central Universities (No.DUT22QN228).

References

- [1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, pages 850–865, 2016.
- [2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *IEEE International Conference on Computer Vision*, pages 6182–6191, 2019.
- [3] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In *European Conference on Computer Vision*, pages 483–498, 2018.
- [4] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *IEEE International Conference on Computer Vision*, pages 10231–10241, 2021.
- [5] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2544–2550, 2010.
- [6] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qihong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: a simplified architecture for visual object tracking. In *European Conference on Computer Vision*, pages 375–392, 2022.
- [7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8126–8135, 2021.
- [8] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [9] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. MixFormer: End-to-end tracking with iterative mixed attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 13608–13618, 2022.
- [10] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate tracking by overlap maximization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4660–4669, 2019.
- [11] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient convolution operators for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6638–6646, 2017.
- [12] Debidatta Dwivedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *IEEE International Conference on Computer Vision*, pages 1301–1310, 2017.
- [13] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Mingzhen Huang, Juehuan Liu, Yong Xu, et al. LaSOT: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129:439–461, 2021.
- [14] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5374–5383, 2019.
- [15] Shenyan Gao, Chunluan Zhou, Chao Ma, Xinggong Wang, and Junsong Yuan. AiATrack: Attention in attention for transformer visual tracking. In *European Conference on Computer Vision*, pages 146–164, 2022.
- [16] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021.
- [17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.
- [18] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2014.
- [19] Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1562–1577, 2019.
- [20] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2011.
- [21] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *IEEE International Conference on Computer Vision*, pages 1125–1134, 2017.
- [22] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Hyung Jin Chang, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, et al. The tenth visual object tracking vot2022 challenge results. In *European Conference on Computer Vision*, pages 431–460. Springer, 2022.
- [23] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019.
- [24] Xin Li, Wenjie Pei, Yaowei Wang, Zhenyu He, Huchuan Lu, and Ming-Hsuan Yang. Self-supervised tracking via target-aware data synthesis. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014.
- [26] Jihao Liu, Boxiao Liu, Hang Zhou, Hongsheng Li, and Yu Liu. TokenMix: Rethinking image mixing for data augmentation in vision transformers. In *European Conference on Computer Vision*, pages 455–471, 2022.

- [27] 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 *IEEE International Conference on Computer Vision*, pages 10012–10022, 2021.
- [28] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool. Transforming model prediction for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8731–8740, 2022.
- [29] Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *IEEE International Conference on Computer Vision*, pages 13444–13454, 2021.
- [30] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for UAV tracking. In *European Conference on Computer Vision*, pages 445–461, 2016.
- [31] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In *European Conference on Computer Vision*, pages 300–317, 2018.
- [32] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302, 2016.
- [33] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.
- [34] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *AAAI Conference on Artificial Intelligence*, volume 36, pages 2071–2081, 2022.
- [35] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam R-CNN: Visual tracking by re-detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6578–6588, 2020.
- [36] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1571–1580, 2021.
- [37] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *IEEE International Conference on Computer Vision*, pages 568–578, 2021.
- [38] Yong Wang, Xian Wei, Xuan Tang, Hao Shen, and Lu Ding. Cnn tracking based on data augmentation. *Knowledge-Based Systems*, 194:105594, 2020.
- [39] Fei Xie, Chunyu Wang, Guangting Wang, Yue Cao, Wankou Yang, and Wenjun Zeng. Correlation-aware deep tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8751–8760, 2022.
- [40] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 12549–12556, 2020.
- [41] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *IEEE International Conference on Computer Vision*, pages 10448–10457, 2021.
- [42] Bin Yan, Xinyu Zhang, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Alpha-Refine: Boosting tracking performance by precise bounding box estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5289–5298, 2021.
- [43] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conference on Computer Vision*, pages 341–357, 2022.
- [44] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.
- [45] Sangdoon Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, and Jinhyung Kim. VideoMix: Rethinking data augmentation for video classification. *arXiv preprint arXiv:2012.03457*, 2020.
- [46] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. MixUp: Beyond empirical risk minimization. In *International Conference on Learning Representation*, 2018.
- [47] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4591–4600, 2019.
- [48] Jianlong Fu Bing Li Weiming Hu Zhipeng Zhang, Houwen Peng. Ocean: Object-aware anchor-free tracking. In *European Conference on Computer Vision*, pages 771–787, 2020.
- [49] Jiawen Zhu, Xin Chen, Dong Wang, Wenda Zhao, and Huchuan Lu. SRRT: Search region regulation tracking. *arXiv preprint arXiv:2207.04438*, 2022.
- [50] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *European Conference on Computer Vision*, pages 101–117, 2018.