Supplementary Material

Zhixiong Pi[1], Weitao Wan[2], Chong Sun[2], Changxin Gao[1], Nong Sang[1], and Chen Li[2]

[1] Key Laboratory of Image Processing and Intelligent Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2] WeChat, Tencent

In this supplementary file, we analyze the implementation details, including the construction of the training sample pairs in the contrastive learning, the selection of the angle modulation strategy, and the selection of the category encoder. Then, we present state-of-the-art comparisons on the sequences with different attributes. Also, we provide more visualization results and a video demo.

1 Implementation details of the network

1.1 Sample pairs in contrastive learning

We elaborate on the negative pair sampling method here. MoCo demonstrates that enough negative pairs is crucial for the representation learning. The MoCo v3 method sets the batch size as 4096, where one sample can construct 4096 negative pairs. With enough negative pairs, MoCo v3 abandons the memory bank. In our experiments, we sample 64 sequences in one batch and randomly select 6 frames in each sequence. In a mini-batch, two samples from any two different sequences are regarded as negative pairs. Therefore, for one sequence, we can construct $6\times6\times63 = 2268$ negative pairs. We do not adopt the memory bank. In our experiments, the contrastive learning is an auxilizry task. Experimental results show that it can work well with relatively small batch size.

Table 1. Performance comparison of two angle modulation strategies.

	OTB100		LaSOT		
	AUC	Prec.	AUC	Prec.	N.Prec.
element-wise product	0.694	0.901	0.578	0.582	0.668
vector addition	0.701	0.910	0.592	0.605	0.685

1.2 Selection of the angle modulation strategy

We test two angle modulation strategies. One is the element-wise product, and another is the vector addition which is the final adopted strategy. In the element-wise product strategy, we replace the vector addition operator as calculating the element-wise product of the backbone feature and modulation feature. We compare the performance of the two strategies on OTB100 and LaSOT datasets. As Table 1 shows, the vector addition achieves better performance. Compared with the element-wise product, vector addition strategy improves 0.7% and 1.4% AUC scores on OTB100 and LaSOT, respectively.

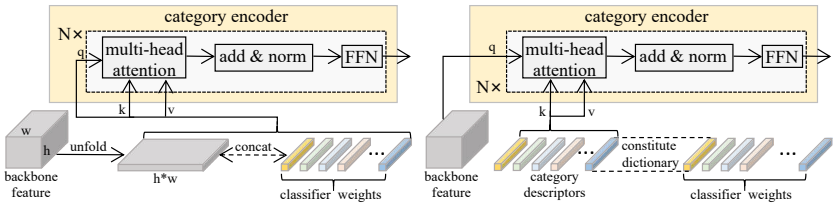


Fig. 1. Two kinds of category encoder. (a) illustrates the concatenation strategy. (b) shows the strategy of taking category classifier weights as the key and value features.

Table 2. Performance comparison of two kinds of category encoder.

	OTB100		LaSOT		
	AUC	Prec.	AUC	Prec.	N.Prec.
concatenation	0.672	0.872	0.556	0.558	0.642
key and value	0.681	0.887	0.568	0.577	0.657

1.3 Selection of the category encoder

We compare two kinds of category encoder. For fair comparison, we add the same category classification branch based on the baseline DiMP18, and do not use the proposed instance-aware module. Then, we add the two category encoders, respectively. We illustrate the two methods in Figure 1. The first method unfold the backbone features in the space dimension, and concatenates the unfolded features and the category classifier weights straightforwardly. The transformer encoder takes the concatenated features as the query, key, and value features. Then, the first $w \times h$ vectors of the output are selected and reshaped to the size of the backbone features as the category-aware features. The second method takes the backbone features as the query features, and the category classifier weights as the key and value features, which is finally adopted in our experiments. The performance of the two methods on OTB100 and LaSOT datasets is shown in Table 2. We annotate the first and the second methods as 'concatenation'

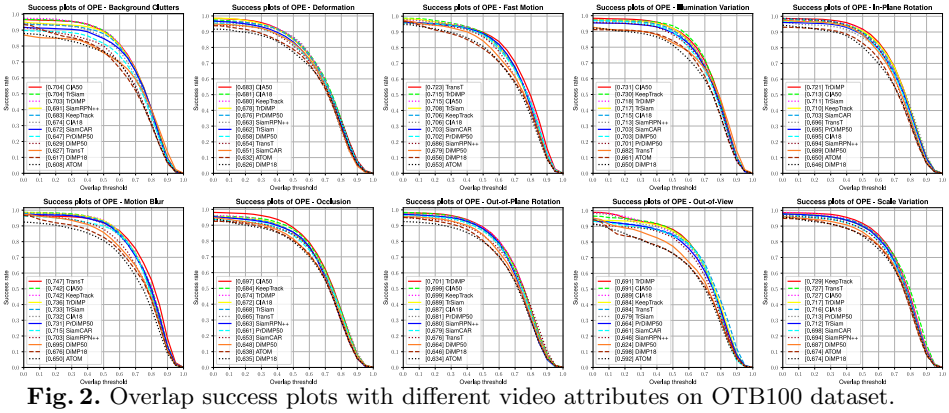


Fig. 2. Overlap success plots with different video attributes on OTB100 dataset.

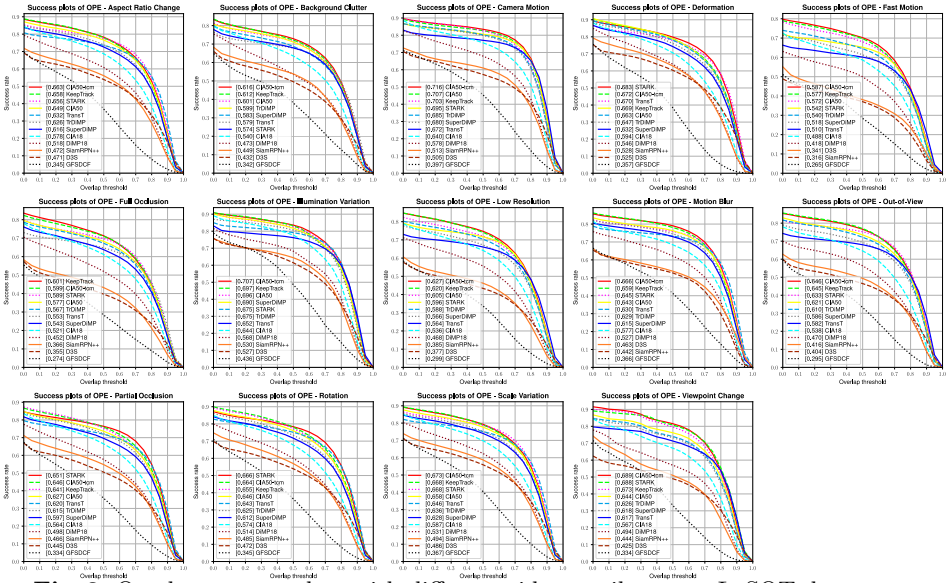


Fig. 3. Overlap success plots with different video attributes on LaSOT dataset.

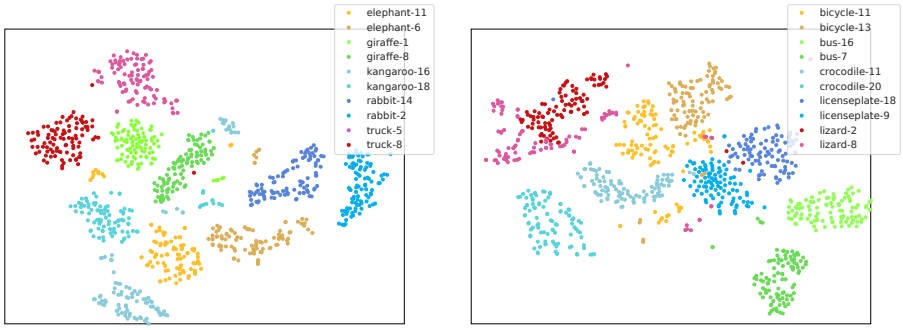
and 'key and value', respectively. It performs better when we take the category classifier weights as the key and value features.

2 More Results on OTB100 Dataset

OTB100 consists of 100 sequences with some different challenge attributes. Figure 2 shows the tracking performance of the state-of-the-art trackers including ATOM, DiMP, TransT, PrDiMP, SiamCAR, KeepTrack, SiamRPN++, TrDiMP, TrSiam, and our CIA trackers, in terms of the overlap success plots on the OTB100 dataset under the different challenging situations. The challenging situations contain 'background clutter', 'deformation', 'fast motion', 'illumination variation', 'in-plane rotation', 'motion blur', 'occlusion', 'out-of-plane rotation', 'out of view', and 'scale variation'. On the sequences with the different attributes, our CIA50 tracker can track the targets precisely and stably.

3 More Results on LaSOT Dataset

LaSOT dataset consists of 1400 video sequences including 280 test videos. The targets are annotated as 70 different categories. The average video length of the LaSOT is about 2500 frames. In Figure 3, we illustrate the tracking performance of some state-of-the-art trackers including GFSDCF, D3S, SiamRPN++, DiMP, TrDiMP, TransT, STARK, KeepTrack, and our CIA trackers, in terms of the overlap success plots, on the sequences from the LaSOT dataset with the different attributes. The video attributes include 'aspect ration change', 'background



(a) known categories (b) unknown categories

Fig. 4. Visualization results of the category classification features from the known and the unknown categories.

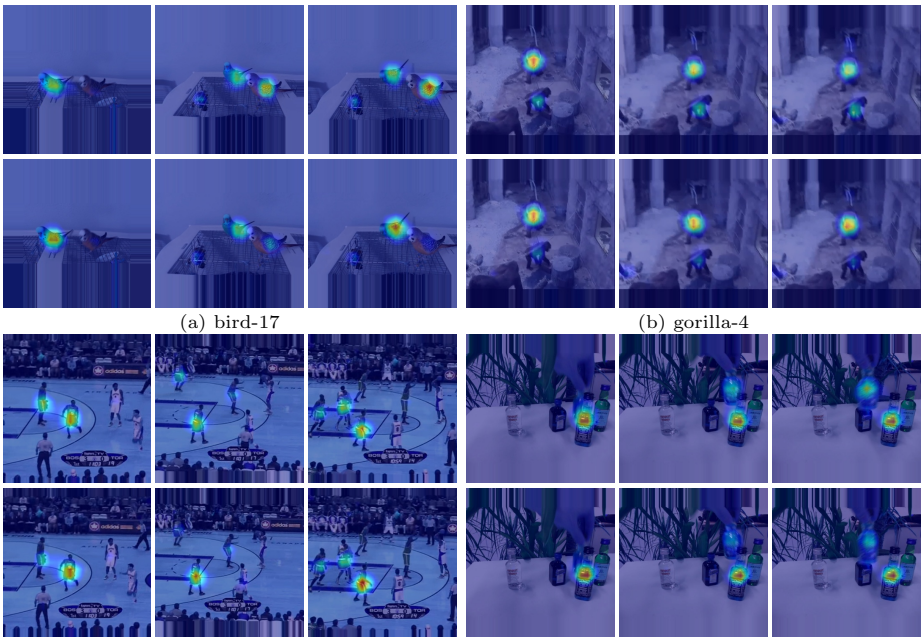
clutter', 'camera motion', 'deformation', 'fast motion', 'full occlusion', 'illumination variation', 'low resolution', 'motion blur', 'out of view', 'partial occlusion', 'rotation', 'scale variation', and 'viewpoint change'. Our CIA50-tcm achieves the best performance under the 'aspect ration change', 'background clutter', 'camera motion', 'fast motion', 'illumination variation', 'low resolution', 'motion blur', 'out of view', 'scale variation', and 'viewpoint change' circumstances.

4 More Visualization Results

In this section, we provide more visualization results of our trackers. First, we visualize the features of our category classification branch. Then, we compare the response maps of our CIA50 tracker and the baseline SuperDiMP.

For comparing the category classification performance on the known and the unknown categories, we re-train our CIA18 model with half of the annotated categories from LaSOT dataset. The category classification features are visualized with t-SNE algorithm in Figure 4. The subfigure (a) and (b) show the feature distributions of some samples from the known categories and the unknown categories, respectively. The visualized known categories are elephant, giraffe, kangaroo, rabbit, and truck. The unknown categories contain bicycle, bus, crocodile, licenseplate, and lizard. Despite being visualized in a low dimension space, we can observe the trend that the features from the same category are close to each other. The samples from the unknown categories are also clustered properly.

We also compare the response maps of our CIA50 tracker and the baseline SuperDiMP in Figure 5. The response maps on the sequences 'Bird-17', 'gorilla-4', 'Basketball', and 'Liquor' are illustrated. The baseline tracker is confused by the similar objects, while our CIA50 tracker can distinguish the real target and the distractors.



(a) bird-17 (b) gorilla-4

(c) Basketball (d) Liquor

Fig. 5. Visualization of response maps. Subfigures (a), (b), (c), and (d) are the response maps on the sequences 'bird-17', 'gorilla-4', 'Basketball', and 'Liquor', respectively. In each subfigure, the upper line shows the response maps generated by the baseline SuperDiMP, and the lower line illustrates the response maps of our CIA50 tracker.

Besides, we provide a video demo in the zip file, showing the bounding box visualization results on several challenging sequences. The compared trackers are SuperDiMP, SiamRPN++, ATOM, TrDiMP, TransT, and our CIA50.