Hierarchical Feature Embedding for Visual Tracking

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

Abstract. Features extracted by existing tracking methods may contain instance- and category-level information. However, it usually occurs that either instance- or category-level information uncontrollably dominates the feature embeddings depending on the training data distribution, since the two types of information are not explicitly modeled. A more favorable way is to produce features that emphasize both types of information in visual tracking. To achieve this, we propose a hierarchical feature embedding model which separately learns the instance and category information, and progressively embeds them.
We develop the instance-aware and category-aware modules that collaborate from different semantic levels to produce discriminative and robust feature embeddings. The instance-aware module concentrates on the instance level in which the inter-video contrastive learning mechanism is adopted to facilitate inter-instance separability and intra-instance compactness. However, it is challenging to force the intra-instance compactness by using instance-level information alone because of the prevailing appearance changes of the instance in visual tracking. To tackle this problem, the category-aware module is employed to summarize high-level category information which remains robust despite instance-level appearance changes. As such, intra-instance compactness can be effectively improved by jointly leveraging the instance- and category-aware modules. Experimental results on various benchmarks demonstrate the proposed method performs favorably against the state-of-the-arts. The code is available on https://github.com/zxgravity/CIA.

Keywords: Instance-level, Category-level, Visual tracking

1 Introduction

Visual object tracking is a fundamental computer vision task, which is widely used in surveillance, automatic drive, and video analysis, to name a few. With the initial target location annotated, the visual tracking algorithms attempt to identify and localize the target object continuously in a video sequence.

* This work was done while Zhixiong Pi was an intern at Tencent.
† Corresponding author

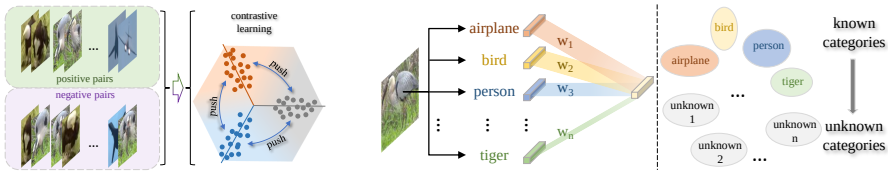


Fig. 1. Illustration of the instance-level and category-level awareness. (a) The model learns the feature embedding that pushes away different instances through contrastive learning. (b) In order to maintain intra-category compactness, the model is trained to produce features from a handful of known categories and generalize to cluster the unknown ones.

Benefiting from the powerful deep neural network, Siamese networks [1,19,18,12] and discriminative modules [6,2,7] have improved the performance significantly. Among these methods, robust target representation learning plays an important role in boosting tracking performance. However, existing tracking methods mainly have two limitations, *i.e.*, the absence of category awareness and the uncontrollable dominance of either instance or category information.

In visual tracking, it is crucial to learn feature embeddings that not only have inter-instance distinction but also have intra-category consistency. Existing tracking methods focus on modeling the discriminative instance features while ignoring the category information modeling. A typical idea of such methods is to learn the instance-specific features by forcing the model to increase the response on the foreground and suppressing that on the background in each frame. However, suppressing the background response may not be effective since many background features are semantically meaningless. Other existing trackers, like DaSiamRPN [43], focus on the distractors by using the hard negative mining method. Similarly, DiMP [2] suppresses the influence of the easy negative samples by masking out the loss from the areas with low response. We argue that it can stabilize target features and improve tracking performance further by modeling category information properly. A simple manner to encode category information in the backbone features is to introduce an additional category loss, leading to the multi-task learning problem. However, there are numerous categories in visual tracking. Explicit category information learning via straightforward multi-task training is sub-optimal, as learned category information is difficult to generalize to the unknown categories with limited annotated categories in the training set.

According to the experimental analysis in current Siamese trackers [19,2], though the category factor is not explicitly considered, the learned features do contain a certain amount of category information. However, the importance of the instance and category information is not properly regularized, which is susceptible to the training data distribution. For example, more category cues are learned when the training data has fewer intra-category distractions, and vice versa. The unconstrained feature learning process increases the overfitting risk making the model to highlight either the instance or the category information.

To overcome these difficulties, we propose to hierarchically embed the instance and category information by explicitly modeling the instance and category

cues in a progressive manner during the feature learning process. We propose the instance-aware and category-aware modules that contribute to producing discriminative and robust feature embeddings from different semantic levels. The instance-aware module, which concentrates on a fine-grained semantic level, employs the proposed video-level contrastive learning mechanism exploiting rich inter-video cues to facilitate inter-instance separability and intra-instance compactness. In this module, we extend the InfoNCE [27] loss to learn an instance feature extractor. Then the instance-level information is effectively embedded into the backbone features through a novel angle modulation strategy which modulates the vectorial angles in the feature space, producing the instance-aware features. To improve intra-instance compactness in spite of instance appearance changes in the sequence, the category-aware module is employed to summarize high-level category information which remains robust although the instance-level appearance can change dramatically. Furthermore, the proposed method can generalize to unknown categories by incorporating a transformer-based dictionary learning approach. We summarize our contributions as follows:

- We propose a novel cross-video training paradigm based on the video-level contrastive learning. The positive and negative training pairs are constructed across videos, which can sufficiently mine the potential of instance distinction of the tracker. Furthermore, we introduce an auxiliary task in visual tracking which employs the momentum contrast to improve the performance.
- We propose the instance-aware module and category-aware module to extract the features with better inter-instance separability and intra-instance compactness. We achieve more understandable feature learning by explicitly encoding the instance and category information with supervision signals.
- We conduct the experiments based on both the ResNet18 and ResNet50 backbones. The ablation studies demonstrate the effectiveness of both the instance- and category-aware modules. The proposed tracker performs favorably against the state-of-the-arts.

2 Related Work

Generic visual object tracking algorithms have achieved remarkable improvement in recent years. Because of its high accuracy and efficiency , the Siamese network based trackers [1,19,18,12,35,29,13] have gained widespread attention. SINT [28] firstly introduces the Siamese network in the visual tracking task. Hereafter, many siamese based trackers are proposed. For locating targets more precisely, The bounding box regression head is integrated in SiamRPN [19], adapting to the scale and ratio changes. SiamMask [35] further improves the localization precision by predicting the target mask. Another way of boosting performance is to exploit the potential of the deep network for the tracking task. With the random shifting augmentation, trackers with very deep networks [18,40] are successfully proposed. In spite of the remarkable improvement brought by the better output formats and the deep backbones, the lack of online updating prevents the trackers from adapting to the target appearance changes, especially in the long-term

sequences. To improve the tracking robustness, ATOM [6] designs a fast online updating method in the inference process to overcome the appearance changes. Then, the online updating method is improved by a meta-learning based updating module [2]. Considering the case of very confusing distractors, the appearance model is unreliable sometimes. To this end, KYS [3] and KeepTrack [24] extract the context information to suppress the confusing distractors. Recently, benefiting from the transformer, the visual tracking accuracy rises significantly. Some trackers integrate the transformer modules into the siamese network to mine the spatial and temporal context information [33,39] or promote the power of the matching head [5].

Despite the competitive performance of the existing trackers, it remains challenging to learn the reliable instance-specific features. Zhu *et al.* [43] improve the instance representation by adding positive and hard negative pairs with data augmentation and using hard negative mining. Wang *et al.* [32] use forward-backward tracking to construct sample pairs and train the tracker with the consistency loss. However, these methods still cannot exploit the instance information sufficiently. Different from them, we follow the idea from He *et al.* [14], that more negative sample pairs can improve the contrastive learning. We construct the sample pairs between the images in the same mini-batch, where the images from the different video sequences. We can construct negative and positive pairs between the different sequences and inside the same sequence. The negative pairs are much more than the positive pairs. Then, we train the tracker via the momentum contrastive learning.

Generic visual tracking needs recognizing and localizing the target annotated in the initial frame without the category information. We observe that the category information, if provided, can stabilize the tracking process. To the best of our knowledge, there are barely any previous works discussing the influence of the category information in generic visual tracking. SA-Siam [13] combines the responses from the semantic network and the appearance network to improve the tracking precision. The semantic network is pretrained on the ImageNet [8] dataset, and the parameters are then fixed, which can be viewed as a model with the category awareness. However, the pretrained semantic network in the SA-Siam model is optimized for the ImageNet classification task, which leads to suboptimal performance in the generic visual tracking task. In contrast, the proposed category-aware module is jointly optimized with the other parts of the tracker for the generic visual tracking task. We demonstrate experimentally that the proposed category-aware module can effectively improve intra-category compactness and stabilize the features of the target instance.

3 Our Approach

In this section, we first briefly summarize the proposed algorithm considering hierarchical feature embeddings. Then, we introduce the details about the instance-aware module which exploits the proposed video-level momentum contrastive learning algorithm to highlight the instance separability ability. Last but not

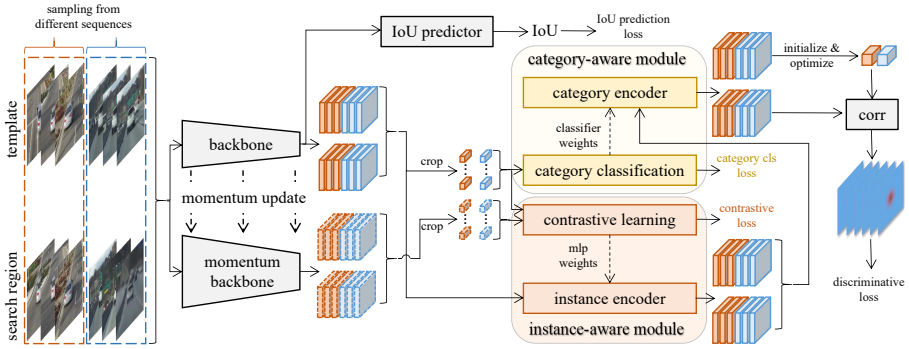


Fig. 2. The pipeline of our approach. The inputs are the frames sampled from video sequences. The features extracted from different sequences are colored with orange and blue, respectively. 'corr' represents the correlation operator. We encode the instance and category information in a progressive manner. The instance encoder integrates instance-aware information into the backbone features. The category encoder then integrates category-aware information into preceding instance-aware features.

least, the category-aware module, which models the stable category information in the feature embedding, is detailed.

3.1 Overview

The pipeline of our algorithm is illustrated in Fig. 2. Inspired by MoCo [14], we introduce a momentum key update mechanism for the contrastive learning formula. Thus, our architecture contains both the prototype and momentum backbones, wherein both backbones share the same architectures containing the template and the search region branches. In each training iteration, we randomly obtain samples from several sequences, and equally split the samples in each sequence as the template and search region samples. The samples are processed by both the prototype and momentum backbones to obtain the intermediate feature representations. Then they are fed into the IoU predictor, as well as the instance-aware module for further computation. As is defined in the previous Siamese trackers [19,43], the feature maps corresponding to the template and search region are named as the template and search region features, respectively. The IoU predictor is implemented following DiMP [2], and it is used to predict the IoU values between the groundtruth and the bounding boxes sampled around.

For the instance-aware module, we crop the intermediate features corresponding to the object regions and generate a group of sample pairs based on the cropped feature maps. In each sequence, we sample three template and three target features. Thus we have $6 \times 6 = 36$ positive sample pairs. Sample pairs from any two different sequences are regarded as negative pairs. These sample pairs are utilized as the inputs of the contrastive learning module which produces the instance-specific features. The instance encoder takes the instance-specific

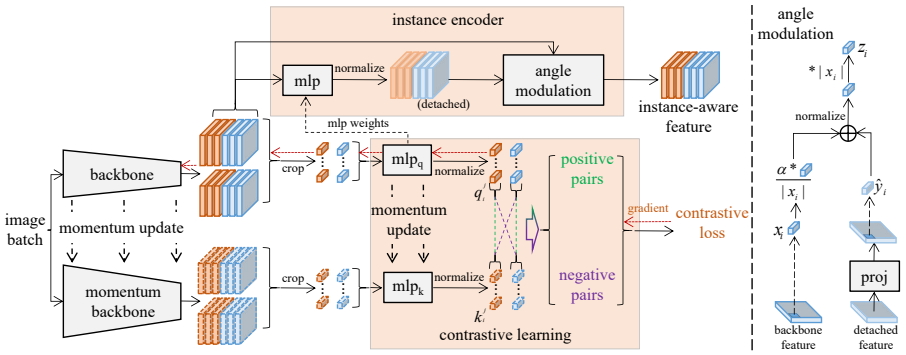


Fig. 3. Our instance-aware module. The module consists of the contrastive learning and the instance encoder. The detail of the angle modulation in the instance encoder is also illustrated at the right of this figure. The orange dash arrows annotate the gradient flows of the contrastive learning. The green and purple dash lines represent the two samples of a sample pair are from the same or the different sequences, respectively.

features and the prototype backbone features as the inputs, and produces the features with instance-level discrimination.

The category-aware module, which takes the computed instance-aware features as inputs, consists of two sub-modules, *i.e.*, the category encoder module and category classification module. Trained from the samples annotated with a handful of known categories, the weight parameters in the category classification module contain rich category encoding information. These weight parameters are regarded as the dictionary atoms and fed into the category encoder module. Then the instance-aware features are modulated with the dictionary, generating the instance- and category-aware target representations. The generated features are then fed into the model optimizer and correlation module following [2]. As shown in Fig. 2, altogether 4 losses are used to optimize the proposed model, including the IoU prediction loss, the category classification loss, the contrastive loss, and the discriminative loss. Among these losses, the definitions of the IoU prediction loss and the discriminative loss are the same as those in DiMP [2]. We employ the cross entropy loss as the category classification loss and extend the InfoNCE [27] loss to make it more suitable for the video-level contrast.

3.2 Instance-Aware Module

In this paper, we explicitly model the instance-level discrimination information considering both the intra- and inter-sequence training samples. A novel instance-aware module is proposed, which consists of the video-level contrastive learning module and the instance encoder. The brief pipeline of the instance-aware module is illustrated in Fig. 3, which will be detailed in this section.

Video-Level Contrastive Learning In the existing tracking algorithms, the positive and negative samples are generated from the same video sequence, wherein the annotated objects are consistently regarded as positive samples. In

such methods, the rich inter-video instance level discrimination information is ignored. Different from the previous implementations, we propose the video-level contrastive learning algorithm, where the annotated object in one video sequence can be regarded as a positive or negative sample in the training pipeline.

The constrasive learning algorithm [14] is a self-supervised method, which learns the feature representations without the need for annotated samples. Given a set of keys $\{k_0, k_1, ..., k_i, ..., k_I\}$ of a dictionary and a query encoder q, we use k_d to denote the matched key of q. The constrasive learning algorithm tries to increase similarity between q and k_d, whilst suppressing the similarity between q and k_i $(i \neq d)$. InfoNCE loss is exploited to achieve this goal:

$$\mathcal{L}_{\text{con}} = -\log \frac{\exp(q^\top k_d/\tau)}{\sum_i \exp(q^\top k_i/\tau)}, \tag{1}$$

where τ is the temperature hyper-parameter.

The original implementation regards samples from the same image as positive training pairs, and regards samples from different images as negative pairs. We extend the contrasive learning method considering the temporal consistency in the video sequence. In our method, we use q_i^j and k_i^j to denote the query and key features for contrastive learning, respectively, where $i \in [1, ..., M]$ denotes sequence index, $j \in [1, ..., N]$ is sample index within each sequence. As is illustrated in Fig. 3, the cropped features in both the upper and lower branches are fed into two multi-layer perceptions (denoted as mlp_q and mlp_k in the figure), each of which consists of two convolutional layers interleaved with a ReLU activation. The outputs of the multi-layer perceptions are normalized to generate q_i^j and k_i^j respectively. Similar to the MOCO method, we also incorporate the momentum update mechanism for the momentum backbone and multi-layer perception mlp_k to ensure the consistent encoders for the keys. Let $\mathcal{B}_q$, $\mathcal{B}_k$, $\mathcal{M}_q$ and $\mathcal{M}_k$ denote the parameters of the backbone, the momentum backbone, mlp_q, and mlp_k , respectively. The momentum update formulas are therefore

$$\begin{aligned} \mathcal{B}_k &\leftarrow \eta\mathcal{B}_k + (1-\eta)\mathcal{B}_q \\ \mathcal{M}_k &\leftarrow \eta\mathcal{M}_k + (1-\eta)\mathcal{M}_q, \end{aligned} \tag{2}$$

where η is the momentum factor.

Based on our extended constrasive learning method, the extended InfoNCE (named as InfoNCE-V in this paper) loss can be rewriten as

$$\mathcal{L}_{\text{con}} = -\log \frac{\sum_i \sum_{j,f} \exp(q_i^j k_i^f/\tau)}{\sum_{i,l}^{i \neq l} \sum_{j,f} \exp(q_i^j k_l^f/\tau)}, \tag{3}$$

which tries to increase the similarity between targets in the same video, whilst suppressing the similarity between targets in different videos. This simple implementation enables our method to simultaneously exploit the cross-video instance training pairs, facilitating the learning of more discriminative features. Compared with InfoNCE, InfoNCE-V allows the positive pair generation between the identical target from different frames in a sequence, supervising our model to cluster samples of the same instance and pushes the different instances away.

Instance Encoder Directly exploiting the features output by mlp_q for target/background classification is suboptimal, as it contains limited category information. The features are susceptible to the drastic appearance changes of the target object. As is described in many recognition papers [9,21], the angle between two feature vectors is crucial to perform instance level discrimination. Inspired by this, we propose the novel angle modulation module to properly embed the instance discrimination information into the backbone features. As illustrated in Fig. 3, the model parameters of mlp_q is copied to mlp, which outputs the instance discriminative convolution feature map sharing the same size with the input feature (feature padding is considered). We use X to denote the backbone feature, then the output feature Y is computed as $Y = \text{mlp}(X)$. Let x_i and y_i denote the feature vector extracted in the i-th position of X and Y respectively, the modulated feature vector z_i is computed as

$$\hat{x}_i = \frac{\alpha * x_i}{|x_i|}, \hat{y}_i = \frac{y_i}{|y_i|} \tag{4}$$

$$z_i = |x_i| * \frac{\hat{x}_i + \hat{y}_i}{|\hat{x}_i + \hat{y}_i|}, \tag{5}$$

where $|.|$ represents the norm of a vector. The learned parameter α controls the modulation strength. More instance discriminative information will be embedded into z_i with a smaller α. By modulating all the elements of the backbone features, we obtain the instance-aware features.

3.3 Category-aware Module

The aforementioned instance-aware features concentrate on the instance discrimination information, which are less robust to the target appearance changes. Further improvement can be achieved by exploiting the categorical information of the instances. Motivated by this, we propose a novel category-aware module on top of the instance-aware module to achieve hierarchical feature embedding. The category-aware module consists of the category classification and the category encoder, the details of which are illustrated in Fig. 4.

Category Classification In the visual tracking task, the so-called classification process means distinguishing the foreground target from the background distractors. To avoid confusion, we define the recognition of the target category as the *category classification* in this paper. In several visual tracking datasets (*e.g.*, LaSOT [10]), each sequence is annotated with one category label. The straightforward way to utilize the category information is to add another branch for category classification, introducing the multi-task training strategy. In our work, we introduce one convolutional block in the category classification module to extract the features, which are then flattened and fed into a fully connected layer for category classification. As is shown in Fig. 4, we adopt the cross entropy loss as the category classification loss $\mathcal{L}_{\text{cls}}$. The parameters of the fully connected layer construct the classifier weights. Assuming there are C known categories, the classifier weights W can be decomposed into C vectors $[w_1, ..., w_c]$, each of

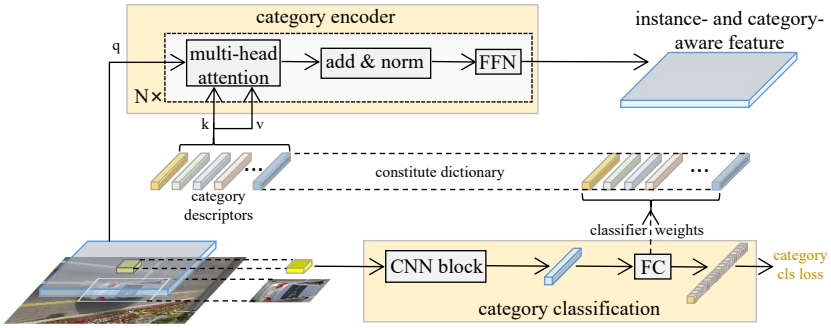


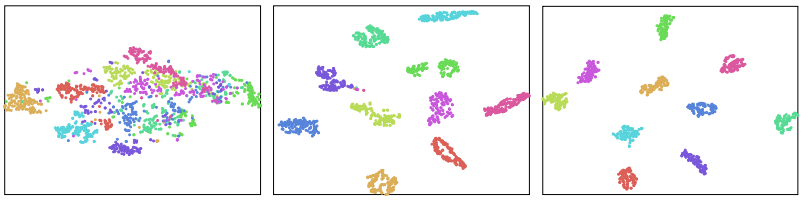
Fig. 4. Our category-aware module. The classifier weights serve as the category descriptors to represent the input features.

which can be viewed as the stable central representation for the corresponding category. During training, the tracker learns to classify the target into these known categories. In the experiments, we further discuss the extreme case where the number of known category C is 0. It is worth noting that directly exploiting multi-task learning mechanism cannot ensure satisfactory performance, as the learned features can hardly be generated to other categories. We further introduce the category encoder module to address this issue.

Category Encoder The category encoder is essentially the concatenated N transformer encoders, which takes the previous computed instance-aware feature as the query feature. The key and value features are both set as W, which is the weight matrix of the category classifier. For an arbitrary instance with known/unknown categories, the transformer encoder outputs target representations encoded via key W, which implicitly performs the dictionary learning process with its column vectors $w_1, ..., w_C$ as the dictionary atoms. Since $w_1, ..., w_C$ are the stable central category representations, the output representations are also stable representations with category information. The instance-aware and category-aware features are combined (via the residual module in the transformer encoder) to obtain the ultimate features for tracking.

3.4 Visualization

We visualize the instance-aware and category-aware features of some targets with the t-SNE [23] algorithm in Fig. 5. The baseline features are extracted by DiMP18. The instance-aware features and the ultimate features are from our CIA18. The instance-aware module produces the instance-aware features which are then enriched by the category-aware module to obtain the ultimate features. The distributions of the baseline, instance-aware, and ultimate features are shown in subfigures (a), (b), and (c) of Fig. 5, respectively. Despite the competitive performance, the instance discrimination of DiMP [2] is unsatisfactory. As subfigure (a) shows, it is hard to split the features of different instances. In contrast, the instance-aware module mines the instance discrimination effectively. After integrating the instance-aware module, the features can be split



(a) baseline features (b) instance-aware features (c) ultimate features

Fig. 5. Visualization results of (a) baseline, (b) instance-aware, and (c) ultimate features with t-SNE [23]. Compared with the baseline features, the instance-aware features have better inter-instance separability. In the ultimate features, the intra-instance compactness is further strengthened.

as subfigure (b) illustrates. The category-aware module can stabilize the target features. As shown in subfigure (c), the intra-instance compactness is further strengthened in the ultimate features, compared to the instance-aware features.

4 Experimental Results

4.1 Implementation Details

Network Architecture. We conduct experiments with the DiMP [2] being the baseline. Our modules are integrated into the baseline trackers DiMP18 [2] and SuperDiMP [2] to obtain our **C**ategory- and **I**nstance-**A**ware (CIA) trackers CIA18 and CIA50. The size of the template image is the same as the search region image. For CIA18 and CIA50, the area of the input image is 5 times and 6 times that of the corresponding target box, respectively. The architecture of the momentum backbone is the same as the feature extraction backbone (*i.e.*, the prototype backbone). The mlp in the instance-aware module consists of a 3×3 convolutional layer, a ReLU activation, and a 1×1 convolutional layer. The projector ('proj' in Fig. 3) in the angle modulation is composed of two 1×1 convolutional layers with a ReLU activation between them. In the category-aware module, the number of the transformer encoder is $N = 3$. The CNN block of the category classification has the same architecture as the layer4 of ResNet.
Training stratagy. The parameters of the proposed instance-aware and category-aware modules are randomly initialized. Then, we train the whole model in an end-to-end manner. The training datasets are GOT-10K [15], TrackingNet [26], COCO [20], and LaSOT [10]. Labeled 70 categories of the LaSOT [10] are the known categories. The samples without category labels are ignored by the category classification. The model is trained on the training video sequences for 50 epochs with an Adam solver with a learning rate decay of 0.2 every 15 epochs. Each epoch includes 2000 iterations. In one batch, we sample images from 64 sequences with 6 random frames in each. There are 4 losses in total, including the contrastive loss $\mathcal{L}_{\text{con}}$, the category classification loss $\mathcal{L}_{\text{cls}}$, the discriminative loss $\mathcal{L}_{\text{dis}}$, and the IoU prediction loss $\mathcal{L}_{\text{iou}}$. The final loss is:

$$\mathcal{L}_{all} = \mathcal{L}_{\text{con}} + \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{dis}} + \mathcal{L}_{\text{iou}}. \tag{6}$$

$\mathcal{L}_{\text{dis}}$ and $\mathcal{L}_{\text{iou}}$ are from [2]. We refer the readers to [2] for mor details. $\mathcal{L}_{\text{con}}$ is our InfoNCE-V loss defined in Eq. 3. $\mathcal{L}_{cls}$ is the cross entropy loss. The parameters of

the momentum backbone and the mlp_k are not updated based on the gradients. Instead, they are initialized by the parameters of the backbone and the mlp_q, and updated via the momentum updating as the Formula 2 lists.

Testing. During the testing phase, we do not need the contrastive learning and category classification parts. The backbone is used to extract the features of the template and the search regions. The template is cropped from the initial frame. We use data augmentation to expand the template, and initialize the template filter according to these augmented images. The template filter is updated during the testing process, like [2] does. For encoding the instance-aware information, we copy the parameters of the mlp_q as those of the mlp in the instance encoder. The padding operation is used for the convolutional layers in the mlp to keep the feature resolution unchanged. The category descriptors remain fixed during the testing phase for encoding the category-aware information. We evaluate our method on various public benchmarks. Our CIA18 and CIA50 run about 43 and 30 fps on one TITAN X GPU, which achieves real-time tracking.

4.2 Ablation Study

We conduct the ablation studies on the OTB100 [37] and the LaSOT [10] datasets. The results of the ablation studies are reported in Table 1. The notations 'CC', 'CE', 'CL', and 'IE' represent category classification, category encoder, contrastive learning, and instance encoder, respectively. If we add the category classification (CC) alone, we obtain minor improvement based on either DiMP18 [2] or SuperDiMP [2] baseline. To utilize the known category information better, we further integrate the category encoder (CE) into the baselines. When adding the category classification (CC) together with the category encoder (CE), we observe a significant improvement. The performance of DiMP18 [2] and SuperDiMP [2] are improved by 3.3% and 1.7% on the LaSOT dataset, respectively. To demonstrate the effectiveness of the instance-aware module, we evaluate the performance of integrating the contrastive learning (CL) and the instance encoder (IE). The contrastive learning (CL) can enrich the instance information, which improves the performance of DiMP18 from 66.0% and 53.5% to 68.4% and 57% AUC scores on the OTB100 and the LaSOT datasets, respectively. Based on the DiMP18 [2] with the contrastive learning, the instance encoder (IE) can

Table 1. The AUC scores (%) of the ablation studies on OTB100 and LaSOT datasets.

baseline	+CC	+CE	+CL	+IE	AUC on OTB100	AUC on LaSOT
DiMP18	-	-	-	-	66.0	53.5
	✓	-	-	-	66.7	54.0
	✓	✓	-	-	68.1	56.8
	-	-	✓	-	68.4	57.0
	-	-	✓	✓	68.6	57.8
	✓	✓	✓	✓	**70.1**	**59.2**
SuperDiMP	-	-	-	-	70.1	63.1
	✓	-	-	-	69.7	63.8
	✓	✓	-	-	70.4	64.8
	-	-	✓	-	70.5	64.3
	-	-	✓	✓	70.8	65.1
	✓	✓	✓	✓	**71.3**	**66.2**

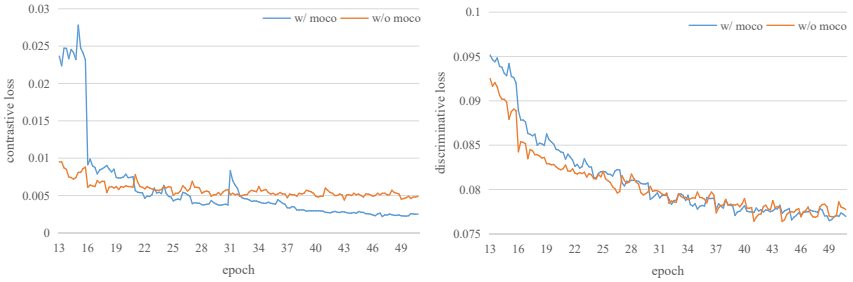


Fig. 6. Contrastive loss and discriminative loss of using momentum update or not.

further improve the AUC scores. The best performance is achieved by using both the complete category-aware module and the instance-aware module. By integrating the proposed modules into DiMP18 [2] and SuperDiMP [2], we obtain the CIA18 and CIA50 trackers, respectively. CIA18 achieves 70.1% and 59.2% AUC scores on the OTB100 and the LaSOT datasets. CIA50 obtains 71.3% and 66.2% AUC scores on the OTB100 and LaSOT datasets, respectively.

We also compare the training strategies of using momentum contrast or not based on our CIA18. Without momentum contrast, we use the same backbone to extract the features of the sample pairs for the contrastive learning. Fig. 6 illustrates the losses of the two training strategies, which validates that momentum contrast facilitates better convergence. Without the momentum contrast, the tracking performance degrades to 65.8%(4.3%↓)/57.4%(1.8%↓) in terms of AUC scores on OTB100/LaSOT datasets. The momentum backbone stabilizes features of queue, prevents from training vibration and leads a better convergence. It is worth noting that we adopt the momentum contrast as an auxiliary task in the supervised learning process. This is the reason why it can work well with relatively small batch size.

Table 2. Influence of the category information.

	CIA18 w/ category	CIA18 w/o category	CIA50 w/ category	CIA50 w/o category
AUC on OTB100	70.1	69.7	71.3	71.2
AUC on LaSOT	59.2	58.5	66.2	65.7

For exploring the influences of the category information, we train our trackers without any known categories, and compare with the performance in Table 2. In this case, the dictionary of the category encoder is initialized randomly and trained with only the discriminative loss. Without any known category, our trackers can also achieve remarkable performance. The known category information can further stabilize the feature extraction and improve the performance.

4.3 State-of-the-art Comparisons

Results on LaSOT [10]. Fig. 7 illustrates the tracking results of the top-performing trackers. On the LaSOT, our CIA50 performs favorably against the state-of-the-arts. After integrating the target candidate matching (tcm) post-processing, like KeepTrack, our CIA50-tcm achieves a 67.6% AUC score, setting a new state-of-the-art record.

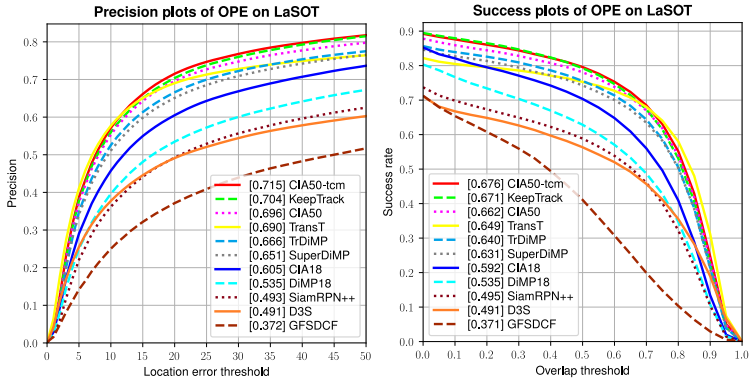


Fig. 7. Precision and overlap success plots on LaSOT dataset.

Table 3. State-of-the-art comparison on TrackingNet test set. The color **red** and **blue** notate the best and the second best result, respectively.

	C-RPN [11]	ATOM [6]	D3S [22]	SiamRPN ++ [18]	DiMP 50 [2]	KYS [3]	SiamFC ++ [38]	PrDiMP 50 [7]	TrDiMP [34]	TransT [5]	CIA 18	CIA 50
Success	66.9	70.3	72.8	73.3	74.0	74.0	75.4	75.8	78.4	**81.4**	74.5	**79.2**
N.Prec.	74.6	77.1	76.8	80.0	80.1	80.0	80.0	81.6	83.3	**86.7**	80.7	**84.5**
Prec.	61.9	64.8	66.4	69.4	68.7	68.8	70.5	70.4	73.1	**80.3**	69.5	**75.1**

Results on TrackingNet [26]. We evaluate the trackers on the TrackingNet dataset with the online evaluation server. The tracking performance are shown in Table 3. Our CIA50 achieves a success score of 79.2% and a normalized precision score of 84.5%. Our CIA18 performs better than the ResNet18 based tracker ATOM with 4.2% success score.

Results on OTB100 [37], UAV123 [25], and NFS [16]. The results on the OTB100, UAV123 and NFS datasets are shown in Table 4. On the OTB100 and NFS datasets, our tracker CIA50 performs the best, surpassing the recent methods KeepTrack, STARK and TrDiMP. Compared with the ResNet18 based trackers like DiMP18, our CIA18 outperforms it on all 3 datasets.

Table 4. State-of-the-art comparisons on OTB100, UAV123, and NFS. The color **red** and **blue** notate the best and the second best result.

	DiMP 18 [2]	TransT [5]	PrDiMP 50 [7]	SiamRPN ++ [18]	Super DiMP [2]	TrSiam [34]	TrDiMP [34]	STARK [39]	Keep Track [24]	CIA 18	CIA 50
OTB [37]	66.0	69.4	69.6	69.6	70.1	70.3	70.8	68.1	**70.9**	70.1	**71.3**
UAV [25]	64.3	**69.1**	68.0	61.3	67.7	67.4	67.5	68.2	**69.7**	66.0	68.9
NFS [16]	61.0	65.7	63.5	50.2	64.8	65.8	**66.5**	66.2	66.4	63.2	**66.7**

Results on GOT-10K [15]. The performance is evaluated on the 180 test video sequences. Table 5 shows the state-of-the-arts comparison results. Our tracker CIA50 achieves a 67.9% AO score, which is 1.8% higher than SuperDiMP.

Results on VOT2020 [17]. The state-of-the-art comparisons on the VOT2020 dataset are shown in Table 6. We compare the bounding box prediction results. Our trackers can achieve competitive performance.

Table 5. State-of-the-art comparison on GOT-10K test set. The color **red** and **blue** notate the best and the second best result, respectively.

	SPM [31]	DiMP 18 [2]	SiamFC ++ [38]	D3S [22]	Ocean [41]	DCFST [42]	Siam RCNN[30]	Super DiMP[2]	TrDiMP [34]	TransT [5]	CIA 18	CIA 50
AO	51.3	57.9	59.5	59.7	61.1	63.8	64.9	66.1	67.1	**67.1**	60.4	**67.9**
$SR_{0.50}$	59.3	67.2	69.5	67.6	72.1	75.3	72.8	77.2	**77.7**	76.8	71.4	**79.0**
$SR_{0.75}$	35.9	44.6	47.9	46.2	47.3	49.8	59.7	58.7	58.3	**60.9**	48.3	**60.3**

Table 6. State-of-the-art comparisons on VOT2020. The color **red** and **blue** notate the best and the second best results.

	SiamFC [1]	ATOM [6]	DiMP50 [2]	UPDT [4]	SuperDiMP [2]	STARK [39]	CIA18	CIA50
EAO(↑)	0.179	0.271	0.274	0.278	0.305	**0.308**	0.278	**0.309**
A(↑)	0.418	0.462	0.457	0.465	0.477	**0.478**	0.462	**0.481**
R(↑)	0.502	0.734	0.740	0.755	**0.786**	**0.799**	0.739	0.782

Results on TNL2K [36]. We evaluate the trackers on the recently proposed TNL2K dataset containing 700 challenging test videos. The comparison results are shown in Fig. 8. Our tracker CIA50 achieves the best performance.

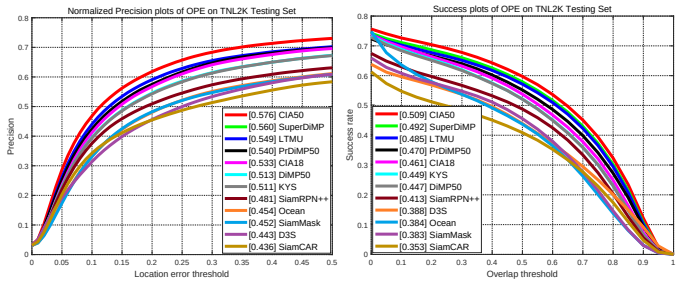


Fig. 8. Normalized precision plots and overlap success plots on TNL2K dataset.

5 Conclusions

In this paper, we propose a novel framework for visual tracking based on instance-level and category-level hierarchical feature embedding. The proposed model extracts deep features by exploiting both intra and inter-video sequences, exploiting richer information for instance discrimination and category generalization. The proposed instance-aware module improves instance discrimination by introducing the contrastive learning method and a novel angel modulation approach to embed the instance information. The category-aware module is developed to generalize the categorical information to unknown categories and enhance categorical consistency, producing stable category descriptors. The instance-aware and category-aware modules are jointly optimized through end-to-end training, achieving the feature embedding that highlights inter-instance separability and intra-instance compactness. Extensive experiments on various benchmarks verify that the proposed method performs favourably against the state-of-the-arts.

6 Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Grant 61433007, Grant 61271328, and Grant 62106149.

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: ECCV Workshops (2016) 2, 3, 14
2. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: ICCV (2019) 2, 4, 5, 6, 9, 10, 11, 12, 13, 14
3. Bhat, G., Danelljan, M., Van Gool, L., Timofte, R.: Know your surroundings: Exploiting scene information for object tracking. In: ECCV (2020) 4, 13
4. Bhat, G., Johnander, J., Danelljan, M., Khan, F.S., Felsberg, M.: Unveiling the power of deep tracking. In: ECCV (2018) 14
5. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu, H.: Transformer tracking. In: CVPR (2021) 4, 13, 14
6. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Atom: Accurate tracking by overlap maximization. In: CVPR (2019) 2, 4, 13, 14
7. Danelljan, M., Gool, L.V., Timofte, R.: Probabilistic regression for visual tracking. In: CVPR (2020) 2, 13
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 4
9. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR (2019) 8
10. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: A high-quality benchmark for large-scale single object tracking. In: CVPR (2019) 8, 10, 11, 12
11. Fan, H., Ling, H.: Siamese cascaded region proposal networks for real-time visual tracking. In: CVPR (2019) 13
12. Guo, D., Wang, J., Cui, Y., Wang, Z., Chen, S.: Siamcar: Siamese fully convolutional classification and regression for visual tracking. In: CVPR (2020) 2, 3
13. He, A., Luo, C., Tian, X., Zeng, W.: A twofold siamese network for real-time object tracking. In: CVPR (2018) 3, 4
14. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020) 4, 5, 7
15. Huang, L., Zhao, X., Huang, K.: 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 (2021). https://doi.org/10.1109/TPAMI.2019.2957464 10, 13
16. Kiani Galoogahi, H., Fagg, A., Huang, C., Ramanan, D., Lucey, S.: Need for speed: A benchmark for higher frame rate object tracking. In: ICCV (2017) 13
17. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.K., Danelljan, M., Zajc, L.Č., Lukežič, A., Drbohlav, O., et al.: The eighth visual object tracking vot2020 challenge results. In: ECCV (2020) 13
18. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: CVPR (2019) 2, 3, 13
19. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: CVPR (2018) 2, 3, 5
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014) 10
21. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: CVPR (2017) 8

22. Lukezic, A., Matas, J., Kristan, M.: D3s - a discriminative single shot segmentation tracker. In: CVPR (2020) 13, 14
23. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11), 2579–2605 (2008) 9, 10
24. Mayer, C., Danelljan, M., Paudel, D.P., Van Gool, L.: Learning target candidate association to keep track of what not to track. In: ICCV (2021) 4, 13
25. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for uav tracking. In: ECCV (2016) 13
26. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: ECCV (2018) 10, 13
27. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018) 3, 6
28. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: CVPR (2016) 3
29. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H.S.: End-to-end representation learning for correlation filter based tracking. In: CVPR (2017) 3
30. Voigtlaender, P., Luiten, J., Torr, P.H., Leibe, B.: Siam r-cnn: Visual tracking by re-detection. In: CVPR (2020) 14
31. Wang, G., Luo, C., Xiong, Z., Zeng, W.: Spm-tracker: Series-parallel matching for real-time visual object tracking. In: CVPR (2019) 14
32. Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W., Li, H.: Unsupervised deep tracking. In: CVPR (2019) 4
33. Wang, N., Zhou, W., Wang, J., Li, H.: Transformer meets tracker: Exploiting temporal context for robust visual tracking. In: CVPR (2021) 4
34. Wang, N., Zhou, W., Wang, J., Li, H.: Transformer meets tracker: Exploiting temporal context for robust visual tracking. In: CVPR (2021) 13, 14
35. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: CVPR (2019) 3
36. Wang, X., Shu, X., Zhang, Z., Jiang, B., Wang, Y., Tian, Y., Wu, F.: Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In: CVPR (2021) 14
37. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions of Pattern Analysis and Machine Intelligence **37**(9), 1834–1848 (2015) 11, 13
38. Xu, Y., Wang, Z., Li, Z., Yuan, Y., Yu, G.: Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In: AAAI (2020) 13, 14
39. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: ICCV (2021) 4, 13, 14
40. Zhang, Z., Peng, H.: Deeper and wider siamese networks for real-time visual tracking. In: CVPR (2019) 3
41. Zhang, Z., Peng, H., Fu, J., Li, B., Hu, W.: Ocean: Object-aware anchor-free tracking. In: ECCV (2020) 14
42. Zheng, L., Tang, M., Chen, Y., Wang, J., Lu, H.: Learning feature embeddings for discriminant model based tracking. In: ECCV (2020) 14
43. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: ECCV (2018) 2, 4, 5