Tracking by Natural Language Specification with Long Short-term Context Decoupling

Ding Ma, Xiangqian Wu
School of Computer Science and Technology, Harbin Institute of Technology
madingcs@hit.edu.cn, xqwu@hit.edu.cn✉

Abstract

The main challenge of Tracking by Natural Language Specification (TNL) is to predict the movement of the target object by giving two heterogeneous information, e.g., one is the static description of the main characteristics of a video contained in the textual query, i.e., long-term context; the other one is an image patch containing the object and its surroundings cropped from the current frame, i.e., the search area. Currently, most methods still struggle with the rationality of using those two information and simply fusing the two. However, the linguistic information contained in the textual query and the visual representation stored in the search area may sometimes be inconsistent, in which case the direct fusion of the two may lead to conflicts. To address this problem, we propose DecoupleTNL, introducing a video clip containing short-term context information into the framework of TNL and exploring a proper way to reduce the impact when visual representation is inconsistent with linguistic information. Concretely, we design two jointly optimized tasks, i.e., short-term context-matching and long-term context-perceiving. The context-matching task aims to gather the dynamic short-term context information in a period, while the context-perceiving task tends to extract the static long-term context information. After that, we design a long short-term modulation module to integrate both context information for accurate tracking. Extensive experiments have been conducted on three tracking benchmark datasets to demonstrate the superiority of DecoupleTNL.

1. Introduction

Tracking by natural language specification (TNL), which aims to localize the specific target referred to by the textual query in a given frame, is a new topic to bridge the two heterogeneous representations of natural language expression and visual content. Therefore, TNL [20, 8, 7, 9, 22] has received more and more attention thanks to the fact that it does not require the manually-specified bounding box to initialize the tracker. The way natural language understanding is incorporated into visual object tracking has many bene-

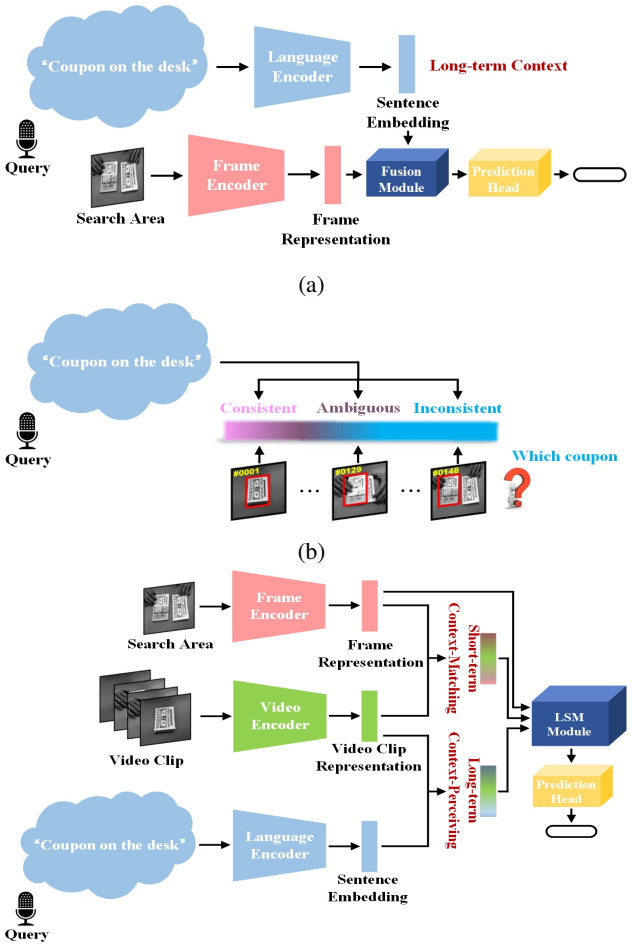

Figure 1: (a) Most TNL methods directly fuse the visual and linguistic contents to localize the specific target object. (b) The textual query describes long-term context information, which may be inconsistent with the visual contents. (c) Our proposed DecoupleTNL framework.

fits, such as breaking the limitation by using a manually-specified bounding box and providing extra scene information from the textual query.

In general, most of the TNL methods share a similar procedure (Figure 1a): encoding visual and linguistic inputs through visual and linguistic components and estimating the location of objects by merging the frame representation and sentence embedding with multi-modal fusion. However, we have found that visual and linguistic content is sometimes inconsistent, in which case a direct fusion of the two may conflict (Figure 1b). To be specific, for the visual part, the model wishes to strengthen the discriminative ability to distinguish the specific target from distractors in the current scene. On the contrary, the linguistic part desires to maximize the representation similarity of objects belonging to the same category. To this end, the inconsistent optimization in these two components limits the development of the current TNL framework in a more accurate way.

Apart from that, most of the previous methods [20, 8, 7, 9, 22] localizing the target may differ in an indirectly manner. In practice, they usually aim to design language-guided candidate matching or selection modules, ignoring the dynamic surroundings through the video flow. To solve this problem, some works have to design an extra module to generate a set of candidates, e.g., region proposals [8] and anchor boxes [9, 32]. Therefore, the performance of these methods is fragile since the predictions are derived from well-designed candidates.

Depending on the above analysis, we propose a self-motivated feature decoupling strategy and a context modulation approach in fusion module design, termed **DecoupleTNL**. DecoupleTNL is implemented with Transformers because the attention module is qualified to establish intra- and inter-modality correspondence for vision and language. The pipeline of the proposed framework is shown in Figure 1c. We decouple the context information into the form of long-term context (i.e., textual query) and short-term context (i.e., video clip). The **Short-term Context-Matching (SCM) branch** aims to guide the video network in capturing the dynamic scene information in a certain period. A contrastive estimation loss is used between the video clip and frame representations, prompting the learning of short-term information. Because the textual query contains information about the entire video, the **Long-term Context-Perceiving (LCP) branch** requires the model to perceive future scenes based on the information of the given video clip. In such a way, the learned representation contains semantic and long-term scene information. In this branch, contrastive learning compares the predicted and "ground truth" at each point through spatial and temporal spans. Then, the long short-term context tokens and frame tokens are fused, and a **Long Short-term Modulation (LSM) module** is utilized to perform feature modulation.

In summary, we draw the contribution of this paper in the following three aspects:

- We analyze the limitations of TNL trackers and propose a novel long short-term context decoupling framework for tracking by NL description. It simultaneously models the long-term and short-term context information for a more robust representation of learning.
- We propose a long short-term modulation module for injecting the long short-term context information into the current frame representation, making our model adaptively perceive dynamic and static surroundings.
- Extensive experiments are conducted on three popular tracking benchmark datasets. The experimental results fully verify the effectiveness of our proposed method for tracking by the natural language specification.

2. Related Work

2.1. Transformer-based Tracking

Motivated by the powerful feature representation capability of vision Transformers, some works integrate the Transformer components into a tracking framework. TransT [3] introduces the Transformer into a Siamese-like framework. TrDiMP [31] explores the temporal contexts across carefully designed parallel branches. These methods relied on postprocessing for box generation. STARK [35] joints the template with the search branch into a single branch. Similarly, DTT [36] builds a discriminative tracker with an encoder-decoder Transformer architecture. ToMP [23] replaces the optimization-based predictor with a newly designed Transformer-based model prediction. Different from TrDiMP [31] and STARK [35] use an encoder-decoder structure to enhance or fuse the features, CSWinTT [27] considers the Transformer as a feature-matching module. To design a simple yet effective end-to-end Transformer-based tracker, MixFormer [4] proposes a mixed attention module for simultaneous feature extraction and information integration. Recently, AiATrack [10] introduces an attention in attention module, which enhances appropriate correlations and suppresses erroneous ones. While Transformer-based tracking by bounding box has achieved competitive performance, they still run the risk of missing the target caused by heavy occlusion and appearance variation.

2.2. Tracking by Natural Language Specification

Currently, only a few works focus on TNL, as it is a new emerging topic. Li *et al.* [20] first give the definition of this task and design two efficient tracking frameworks with a textual query named TNL, which utilizes a textual kernel to search on the visual features. Inspired by the one-stage

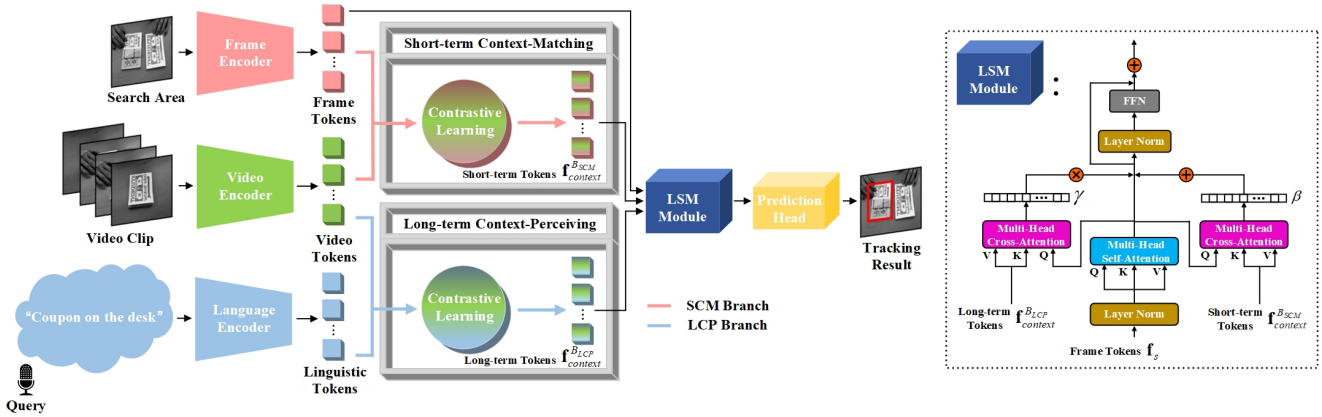

Figure 2: An overview of the proposed tracking framework. There are three feature encoders used to extract the frame, video, and linguistic tokens, respectively. Then, we design two branches, i.e., the **Short-term Context-Matching (SCM) branch** and **Long-term Context-Perceiving (LCP) branch**, gathering meaningful long short-term context information with a short-term context-matching task and a long-term context-perceiving task. After that, we associate the frame tokens and the learned long short-term tokens representations through a **Long Short-term Modulation (LSM) module**. In the end, there is a prediction head for localizing the target object.

regression tracker SiamRPN [19], NLRPN [8] adds a natural language region proposal network to output the proposals jointly. To speed up TNL trackers, RTNL [7] devises a language network conditioned on the regions proposed by a visual network. Similarly, SNLT [9] proposes a universal Siamese Natural Language Region Proposal Network. To mutually promote the learning of two heterogeneous features, CapsuleTNL [22] introduces two routing modules to facilitate the cluster of linguistic and visual representations through Capsule Network. To make aware the global scene , Wang *et al.* [32] introduce a method based on an adaptive local-global-search scheme. Different from these methods, we decouple the context information into long-term and short-term manners to further improve the aggregation of linguistic and visual information.

2.3. Self-supervised Video Learning

Recently, contrastive learning has gained more attention in image representation learning [24, 12], which achieves almost a similar performance to the supervised counterpart. In view of its advances, researchers have introduced the idea of contrastive learning to the video domain [28, 30], where clips of the same domains are pulled together, and clips of different domains are pushed apart. To separate the motion information from raw RGB, Huang *et al.* [14] decouple the motion supervision from the context bias in the pretext task. For the multi-modality of videos, many works investigate mutual supervision across modalities to capture high-quality video representation. For instance, they consider the consistency between videos and the source of audio [17, 1, 34] as supervision for contrastive learning. Our work is also motivated by contrastive video representation learning, where the video representation is explicitly decoupled into static and dynamic contextual information.

3. Proposed Method

3.1. Overall Framework

As depicted in Figure 2, given the search area, a video clip, and the textual query as inputs, we first use three encoders to generate frame, video, and sentence embedding. Then, in the SCM branch, a context-matching task is designed for capturing the short-term and dynamic context representation. Meanwhile, in the LCP branch, a context-perceiving task is devised for gathering the long-term and static context representation. After that, we introduce an LSM module, which plays the roles of both frame feature refinement and multi-modal fusion. Different from the previous methods, we get rid of the candidate box selection and optimization by localizing the referred region with a simple and elegant prediction head. In the following subsections, we elaborate on the details of each component below.

3.2. Encoders

Frame and Video Encoder. The frame encoder comprises a convolutional backbone network and a transformer encoder layer. We select the commonly used ResNet [9, 32, 7] as the backbone network. We apply a 1×1 convolutional layer to reduce the channels of $Res3d$ and $Res4f$ from 512 and 1024 to 256, respectively. The transformer encoder layer comprises a multi-head self-attention layer and an feed-forward network (FFN). Specifically, the multi-head attention layer contains eight heads, and the FFN includes two FC layers with a ReLU activation layer. By feeding the video frame $\mathbf{f} \in \mathbb{R}^{3\times W\times H}$ into the backbone network, we gather the fourth convolutional blocks' feature maps $\mathbf{f'} \in \mathbb{R}^{C\times w\times h}$ where $w = \frac{W}{16}$ and $h = \frac{H}{16}$. Then, the frame tokens are obtained by flattening $\mathbf{f'}$ to $\mathbf{f}_s \in \mathbb{R}^{C\times N_s}$ where $N_s = w \times h$. We use a sine function to generate spatial

positional encoding and add them with the query and key embedding.

Given a set of historical search areas with the temporal scope of T, we first concatenate the local representations among the tokens of those historical search areas along the number of tokens axis. Then, we employ an FFN to obtain the video clip representations $\mathbf{f}_v \in \mathbb{R}^{C\times N_v}$, which is complementary to the global expression of a textual query.

Language Encoder. The language encoder includes a token embedding layer and a linguistic transformer. The linguistic transformer is stacked with 12 transformer encoder layers. Each word is first converted to a one-hot vector. Followed by [5], we generate a linguistic token corresponding to a one-hot vector. The linguistic embedding is denoted as $\mathbf{f}_l \in \mathbb{R}^{C_l\times N_l}$, where C_l and N_l are the channel dimension and number of linguistic tokens, respectively.

3.3. Short-term Context-Matching Branch

In the short-term context, there is a strong spatial-temporal relationship between the scenes in consecutive frames. We present a context-matching task to capture such information. The context-matching process is shown as the **SCM Branch** in Figure 2. To match the size of frame tokens, we reshape $\mathbf{f}_v \in \mathbb{R}^{C\times N_v}$ to $\mathbf{f}_{context}^{B_{SCM}} \in \mathbb{R}^{C\times N_s}$ with an FC layer followed by a ReLU activation layer. Mathematically, abbreviating $\mathbf{f}_s$ and $\mathbf{f}_{context}^{B_{SCM}}$ as $\mathbf{z}$ and $\mathbf{x}$ respectively, the context-matching task is optimized by an InfoNCE loss [24]:

$$\mathcal{L}_{short} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\cos(\mathbf{z}_i,\mathbf{x}_i)/\tau)}{\sum_{k=1}^{N}\exp(\cos(\mathbf{z}_k,\mathbf{x}_i)/\tau)}, \quad (1)$$

where N is the number of samples in a mini-batch, $\cos(\cdot)$ is the cosine similarity between $\mathbf{f}_s$ and $\mathbf{f}_{context}^{B_{SCM}}$, and τ is a temperature parameter. The supervision of the context-matching task is the mean value of the video clip. The loss function pulls video clips and frames from the short-term context information together.

3.4. Long-term Context-Perceiving Branch

Static long-term context information is more sensitive to discriminative characteristics than dynamic short-term context information. Therefore, we devise a context-perceiving task that utilizes the current video clip to predict the future scene provided by the textual query. Here, we use the linguistic tokens extracted from the text query as the supervision. The context-perceiving process is shown as **LCP Branch** in Figure 2. Similar to the context-matching task, we obtain the compressed linguistic tokens $\mathbf{f}_l^{'}$ to match the size of $\mathbf{f}_s$, and $\mathbf{f}_{context}^{B_{LCP}}$ is generated by the same operation as $\mathbf{f}_{context}^{B_{SCM}}$ with another FC layer. Then, the context-perceiving task is optimized by another InfoNCE loss $\mathcal{L}_{long}$ with the same form as $\mathcal{L}_{short}$. The only difference is that $\mathbf{z}$ and $\mathbf{x}$ denote $\mathbf{f}_l^{'}$ and $\mathbf{f}_{context}^{B_{LCP}}$ in $\mathcal{L}_{long}$, respectively. In such a way, $\mathcal{L}_{long}$ leads to learning the static and discriminative information throughout the whole video.

3.5. Long Short-term Modulation Module

Given the short-term tokens $\mathbf{f}_{context}^{B_{SCM}} \in \mathbb{R}^{C\times N_s}$ extracted from the SCM branch and the long-term tokens $\mathbf{f}_{context}^{B_{LCP}} \in \mathbb{R}^{C\times N_s}$ of the LCP branch, we apply the LSM module to enhance $\mathbf{f}_s$ with comprehensive context information by performing layer modulation. The architecture of the **LSM module** is illustrated in Figure 2. Particularly, the long short-term context representations are injected into the frame tokens by adding two extra multi-head cross-attention (MHCA) modules between the MHSA module and the FFN of the original frame encoder layer. Take the long-term part as an example, the outputs of the MHSA module are regarded as the *query* embedding of the MHCA module. In such a manner, the language prompt tokens derived from this MHCA module are the aggregation of frame features with the main characteristics throughout the video. To adaptively control the amount of long short-term information, we encode $\mathbf{f}_{context}^{B_{SCM}}$ and $\mathbf{f}_{context}^{B_{LCP}}$ to modulate the frame tokens by scaling and shifting. To be specific, the static context information stored in $\mathbf{f}_{context}^{B_{LCP}}$ is projected into a scaling vector γ, and the dynamic context information contained in $\mathbf{f}_{context}^{B_{SCM}}$ is projected into a shifting vector β with two MLPs:

$$\gamma = \tanh(W_\gamma \mathbf{f}_{context}^{B_{LCP}} + b_\gamma), \quad (2)$$

$$\beta = \tanh(W_\beta \mathbf{f}_{context}^{B_{SCM}} + b_\beta), \quad (3)$$

where W_γ, W_β, b_γ, and b_γ are learnable parameters. After that, $\mathbf{f}_s$ is refined with the two modulation vectors:

$$\mathbf{f}_s^{'} = F(\mathbf{f}_s)\odot\gamma + \beta, \quad (4)$$

where $\odot$ represents Hadamard product. F consists of a 1×1 convolutional layer followed by an instance normalization layer. Then, $\mathbf{f}_s^{'}$ is fed into the following FFN to obtain the input of the next layers. We stack several LSM modules further to refine the frame representation in such a manner. In the end, we denote the output of the LSM module as $\mathbf{f}^* \in \mathbb{R}^{256\times256}$. Notably, we add learnable position encodings as the input of each transformer encoder layer to retain the positional information.

Deep insights into the long short-term context decoupling strategy. There may be inconsistencies between the frame representation and the sentence embedding because the sentence embedding contains the global information of the video. Such inconsistent information may lead to drift when the target occurs large appearance variation, occlusion, disappears and then reappears. To address this issue, we propose a context decoupling strategy to model the short-term and long-term context information via short-term context-matching and long-term context-perceiving

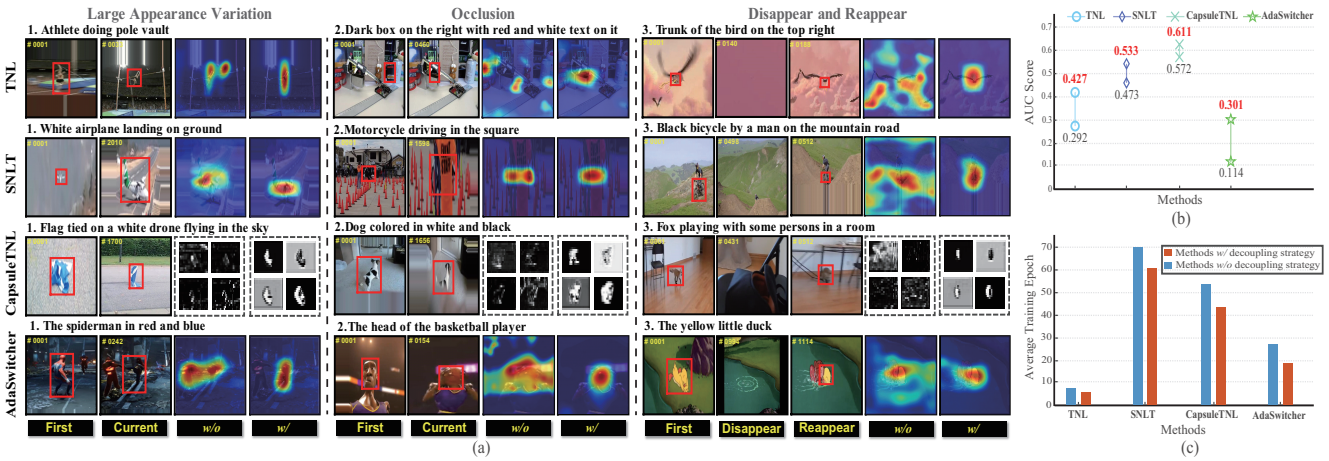

Figure 3: (a) Visual comparison between methods *w/* and *w/o* long short-term context decoupling strategy. (b) Performance gains when equipping with the proposed strategy. (c) Training loss *w/* and *w/o* the proposed strategy.

tasks. On top of that, it ensures that relevant can be gathered to facilitate accurate tracking. As shown in Figure 3, we add the proposed context decoupling strategy to existing TNL trackers, including TNL [20], SNLT [9], CapsuleTNL [22], and AdaSwitcher [32], initializing with only the natural language. Here, we strictly follow the implementation settings mentioned in their literature. Thanks to the adaptive context information learned by the context decoupling strategy, the baseline networks (i.e., TNL, SNLT, CapsuleTNL, and AdaSwitcher) can generate the feature map (e.g., TNL, SNLT, and AdaSwitcher) or pose matrix of capsules (e.g., CapsuleTNL) more precisely. More visible in Figure 3(b,c), benefit from the proposed context decoupling strategy, these methods achieve better performance and converge faster.

3.6. Prediction Head

Similar to previous works [11, 29, 36], the prediction head of our model consists of two heads, one for predicting the foreground or background probability for each location and the other one for computing the target bounding box. Mathematically, these two predictions can be expressed as:

$$\mathbf{P}^{cls} = \varphi^{cls}(\mathbf{f}^{*}), \mathbf{P}^{reg} = \varphi^{reg}(\mathbf{f}^{*}), \tag{5}$$

where $\varphi^{cls}(\cdot)$ and $\varphi^{reg}(\cdot)$ denote the FFNs for classification and regression, respectively. The output of $\mathbf{P}^{cls}$ is a 2-D vector indicating the foreground and background scores of the corresponding location. Differently, $\mathbf{P}^{reg}$ outputs a 4-D vector that represents the distances from the corresponding location to the four sides of the bounding box. Then, the procedure of predicting the bounding box is similar to [36].

3.7. Optimization

Objective Function. For context-matching and context-perceiving tasks, we linearly combine these two losses:

$$\mathcal{L}_{context} = (1-\lambda)\mathcal{L}_{short} + \lambda\mathcal{L}_{long}, \tag{6}$$

where the λ is a scalar hyper-parameter. Given the long short-term enhanced features, the prediction head is used to output binary classification and regression results. Especially, the binary cross-entropy loss is utilized for classification, which is denoted as:

$$\mathcal{L}_{cls} = -\sum_{j}[y_j \log(p_j) + (1-y_j)\log(1-p_j)], \tag{7}$$

where y_j is the label of the $j-th$ sample and p_j indicates the probability that the prediction belongs to the foreground.

For regression, we employ the generalized IoU loss $\mathcal{L}_{GIoU}$ [26]. The total losses are expressed as follows:

$$\mathcal{L} = \alpha{\cdot}\mathcal{L}_{context} + (\mathcal{L}_{cls} + \mathcal{L}_{GIoU}), \tag{8}$$

where α is the regularization parameters in our experiments.

Training. Following the SNLT tracker [9], we collect the images and queries from VisualGenome [18], MSCOCO [21], and Youtube-BoundingBox [25], joint with the training split of LaSOT [6] and OTB-lang [20] for training the proposed method. Notably, when testing on the TNL2k dataset [32], we only used the TNL2k training set.

Initialization. *Tracking by Natural Language and BBox* (*NL + Box*): we initialize the tracker based on language and BBox. We perform data augmentation by constructing 100 training samples and copying the first frame five times to form the video clip. We use these samples to optimize the parameters of the whole model for 50 iterations. *Tracking by Natural Language only* (*NL only*): we first feed the linguistic tokens into the prediction head to locate the target object. Based on the predicted location, we initialize the tracker using the same process as *NL + Box*.

Table 1: Comparisons on OTB-lang, LaSOT, and TNL2k datasets. Here, the AUC and precision (P) scores are reported for the three datasets, respectively.

Method	OTB-lang [20]		LaSOT [6]		TNL2k [32]	
	AUC	**P**	**AUC**	**P**	**AUC**	**P**
TNL [20]	0.252	0.292	-	-	-	-
RTNL [7]	0.542	0.784	0.284	0.281	-	-
SNLT [9]	-	-	0.473	0.478	-	-
AdaSwitcher [32]	0.191	0.242	0.511	0.493	0.114	0.064
CapsuleTNL [22]	0.672	0.886	0.572	0.588	-	-
TNL [20]	0.553	0.723	-	-	-	-
NLRPN [8]	0.671	0.811	0.500	0.563	-	-
RTNL [7]	0.613	0.793	0.353	0.353	0.250	0.272
STNL [9]	0.666	0.848	0.540	0.574	0.248	0.269
AdaSwitcher [32]	0.682	0.881	0.512	0.552	0.417	0.420
CapsuleTNL [22]	0.711	0.924	0.615	0.633	-	-
Ours (NL only)	0.695	0.928	0.649	0.671	0.407	0.400
Ours (NL + Box)	**0.738**	**0.948**	**0.712**	**0.753**	**0.567**	**0.560**

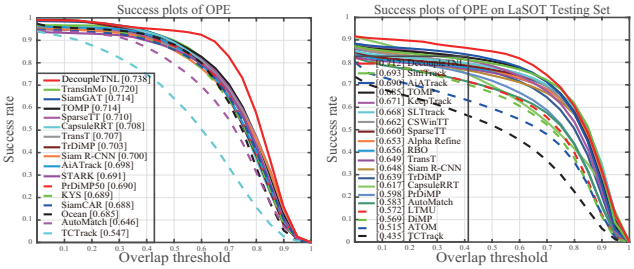

Figure 4: Success plots on OTB-lang and LaSOT, respectively.

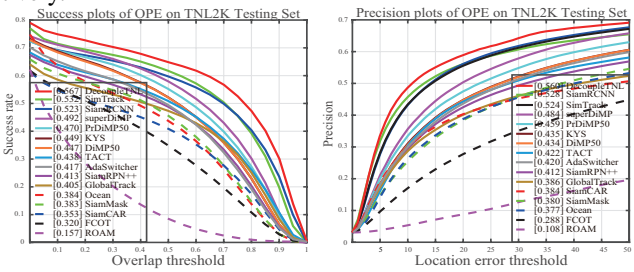

Figure 5: Success and precision plots on TNL2k, respectively.

Inference. During the inference phase, we first compute the prediction of $\mathbf{P}^{cls}$ and $\mathbf{P}^{reg}$. Depending on the highest foreground score, we predict the bounding box as in [36].

Updating. When the current frame is predicted, we only replace the search area of the earliest frame in the video clip with its corresponding search area.

4. Experiments

4.1. Implementation Details

Our model is implemented in Pytorch and runs on the hardware with Intel(R) 10700k CPU and Nvidia 3090 GPU. The input of both networks is resized to 256×256. The maximum length of the textual query is set as 40. If the length of the textual query is shorter than 40, we pad empty tokens after [SEP] token to make it equal to 40. Otherwise, we cut off the language query if its length is longer than 38. We stack four LSM modules in our tracker. The whole is trained for 47 epochs with 1,000 iterations per epoch. The ADAM optimizer [16] is also used with an initial learning rate of 1e-4, and sets a decay factor 0.2 per 10 epochs. During the online updating, we decreased the learning rates to 2e-7. Our DecoupleTNL runs at 32 FPS.

4.2. Comparisons to NL-based Trackers

We evaluate trackers under two different initialization and tracking settings: *NL only* (light gray part in Table 1) and *NL + Box* (dark gray part in Table 1).

OTB-lang [20]. As shown in Table 1, we can find that DecoupleTNL (*NL + Box*) achieves the best performance on OTB-lang [20], i.e., 0.738/0.948 on the success/precision score, respectively. We notice that DecoupleTNL (*NL only*) achieves comparable performance against the trackers initialized with *NL + Box*, only slightly worse than DecoupleTNL (*NL + Box*). We attribute such a favorable performance of DecoupleTNL to the decoupling strategy.

LaSOT [6]. As illustrated in Table 1, on the larger dataset LaSOT, there are a total of six trackers evaluated for this benchmark. Our DecoupleTNL (*NL + Box*) can achieve 0.712/0.753 on the two metrics, which surpasses RTNL, NLRPN, and SNLT by a large margin. The experiments on LaSOT validate the effectiveness of our tracker.

TNL2k [32]. As depicted in Table 1, on the more challenging dataset TNL2k, AdaSwitcher [32] estimates the target state with a local tracking algorithm and global grounding module and achieves 0.417/0.420 on success/precision plots, respectively. When we decouple the context information in a long-term and short-term manner, we achieve 0.567/0.560 on the TNL2k dataset. Although our method has achieved good competitive results on TNL2k, it still cannot solve the problem of switching target objects during tracking, such as "The player who controls the ball", and "The orange calabash which can become a baby", etc.

4.3. Comparisons to State-of-the-art Trackers

OTB-lang [20]. In addition to the above comparison with the NL-based trackers, we also evaluate the proposed DecoupleTNL with other state-of-the-art trackers on the OTB-lang benchmark. OTB-lang is an extended version of OTB-100 [33], where each sequence is annotated with a textual query. As shown in Figure 4, the proposed DecoupleTNL achieves the advanced performance of 0.738, outperforming all the compared trackers.

LaSOT [6]. To further evaluate the proposed tracking

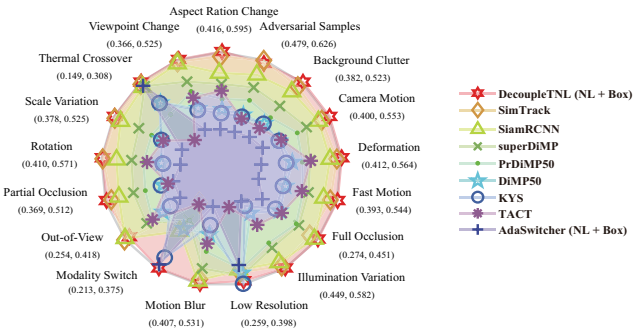

Figure 6: AUC scores of different attributes on TNL2k.

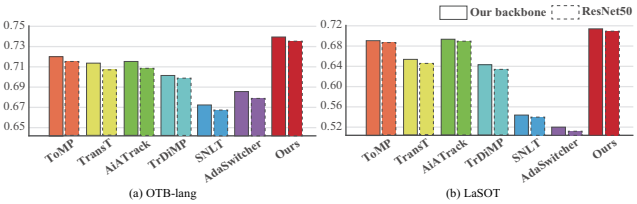

Figure 7: Comparison of different backbones.

method, we conduct experiments on the challenging benchmark LaSOT [6]. We report the results in Figure 4, and the proposed tracker DecoupleTNL achieves the best performance among all the compared trackers. Compared with the recently released trackers SimTrack [2], AiATrack [10], TOMP [23], SLTtrack [15], DecoupleTNL performs better. Notably, we obtain the best AUC score of 0.712.

TNL2k [32]. It should be noted that adversarial samples and thermal images are introduced to verify the robustness of trackers. We utilize the bounding box and natural language query initialization to evaluate all the compared methods on 700 video sequences. As illustrated in Figure 5, DecoupleTNL achieves the best AUC score of 0.567, exceeding 3.5% and 3.6% on success and precision metrics, compared to the second-best tracker SimTrack.

In addition, as shown in Figure 6, we select the top nine performance trackers for the comparison of 17 attributes. The proposed DecoupleTNL outperforms all the comparison trackers, which demonstrates that the proposed tracking paradigm is capable of handling different complex scenes.

Since our backbone is slightly different from other state-of-the-art trackers (e.g., ToMP [23], TransT [3], AiATrack [10], TrDiMP [31], SNLT [9], and AdaSwitcher [32]), we conducted an experiment to verify the impact of backbone on model performance (see Figure 7). When these trackers use the same backbone as ours, i.e., ResNet50 [13]+our transformer layer, their results are lower than our method. When we replace the backbone with ResNet50 and only utilize a 1×1 convolutional layer to match the output dimensions, our results are still better than those methods. These results demonstrate that our method is able to achieve state-of-the-art results while using the same backbone.

Table 2: Ablation study of each component, where $\mathbf{f}_s$, SCM, LCP, LSM and P are short for frame tokens, SCM branch, LCP branch, LSM module and prediction head, respectively. Here, the AUC scores are reported for the three datasets, respectively. $(\cdot)$ indicates the results initialized with (*NL only*).

	SCM	LCP	LSM	P	OTB-lang [20]	LaSOT [6]	TNL2k [32]
$\mathbf{f}_s$				✓	0.535(0.406)	0.427(0.364)	0.318(0.288)
$\mathbf{f}_s$	✓			✓	0.674(0.625)	0.605(0.544)	0.367(0.305)
$\mathbf{f}_s$		✓		✓	0.665(0.617)	0.620(0.559)	0.378(0.329)
$\mathbf{f}_s$	✓	✓	✓	✓	**0.738(0.695)**	**0.712(0.649)**	**0.567(0.407)**

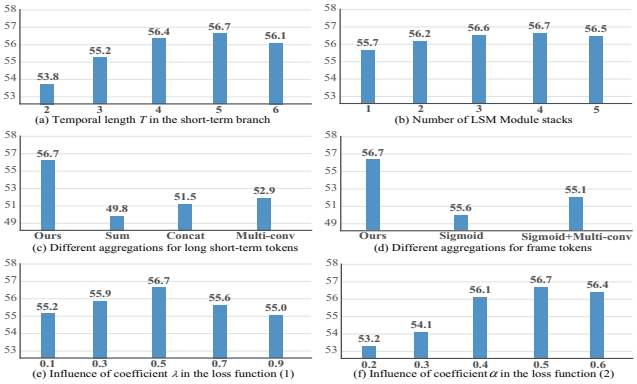

Figure 8: Ablation studies about aggregations and hyper-parameters, measured by AUC score on TNL2k.

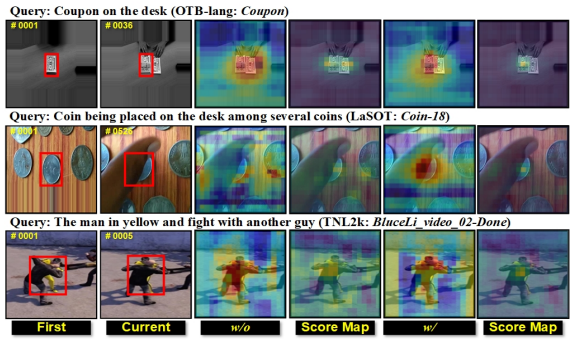

Figure 9: Ablation studies about context decoupling.

4.4. Ablation Experiments

To verify the effectiveness of DecoupleTNL, we conduct ablation experiments on three benchmarks. The comparison results are illustrated in Table 2.

Efficient of each component. Table 2 illustrates the efficacy of each component on three benchmark datasets. Notably, the performance of both branches has its own advantages, demonstrating the necessity of long short-term context decoupling strategy. Figure 9 visualizes the tokens of both branches, where it shows the score map differences with and without dual branches. SCM branch predicts better when the target undergoes dynamic changes. In contrast, the LCP branch consults representative exemplars from the textual query and consistently outputs high confidence for

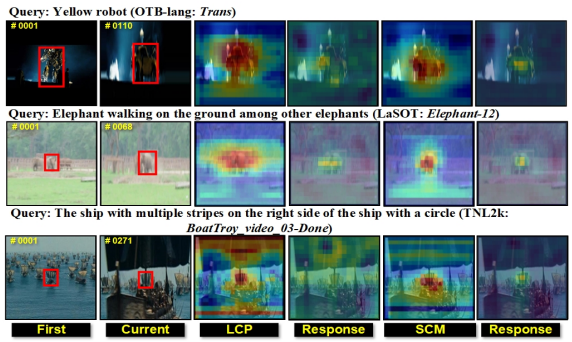

Figure 10: Ablation studies about SCM branch.

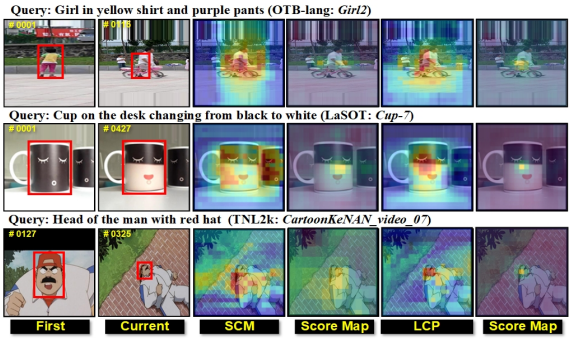

Figure 11: Ablation studies about LCP branch.

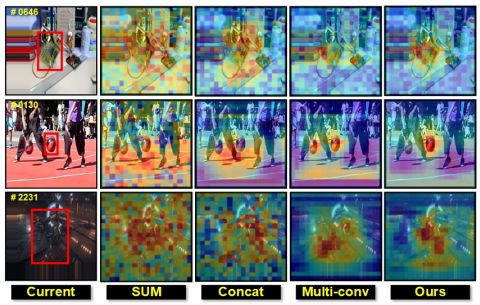

Figure 12: Ablation studies about LSM module.

the case of large movement or reappear.

Ablations about SCM branch. As illustrated in Figure 10, we show some visual examples. Obviously, the effect of the SCM branch is more pronounced when the target appearance changes dynamically. As shown in Figure 8(a), we study the influence of temporal scope T for context-matching and explore that 5 is a proper choice for the SCM branch. The reason behind this may be that too long temporal scope would bring noises, while the too short temporal scope is insufficient to gather meaningful context information.

Ablations about LCP branch. LCP branch characterizes the change of the target and its surroundings throughout the video. As depicted in Figure 11, it can be seen that the LCP branch can effectively improve the performance of the tracker. The superiority of the LCP branch is that when the target has a large displacement or reappears after being occluded, the model can accurately locate the target.

Ablations about LSM module. As shown in Figure 12, models capitalizing on the LSM module achieve better accuracy, which demonstrates the advantage of the proposed fusion strategy. Besides, as depicted in Figure 8(b), we stack four LSM modules in our tracker. Apart from this, without the LSM module, directly fusing the outputs of two branches (e.g., using the sum, concat, and multi-conv operation, see Figure 8(c)) only gains limited improvements, while the proposed LSM module can further improve the tracking performance. Compared with the whole scene information contained in the long-term tokens and the previous context information contained in the short-term tokens, only frame tokens contain information about the current frame. Therefore, we need to make full use of frame tokens. In Figure 8(d), we find that sigmoid or sigmoid + multi-conv may ignore some information about the current frame, resulting in a decrease in performance.

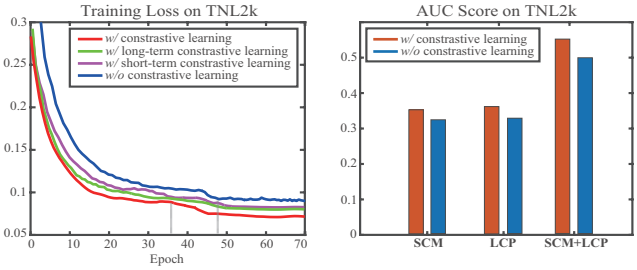

Figure 13: Ablation studies about contrastive learning.

Ablations about hyper-parameters. Figure 8(e,f) studies the coefficient λ and α for the loss functions. We find that $\lambda = 0.5$ and $\alpha = 0.5$ are proper choices.

Ablations about contrastive learning. In Figure 13, we perform an analysis on contrastive learning for *NL + Box*, where the variants without constructive learning mean that we concatenate the tokens of this branch with the frame tokens. Obviously, contrastive learning can not only reduce the training time but also improve the model performance.

5. Conclusions

This paper proposes DecoupleTNL, decoupling the context information into short-term and long-term forms. The short-term context information is gathered by a context-matching task, while the long-term information is captured by a context-perceiving task. We embed these two types of context information into the visual tracking framework, fusing them with the frame representation to obtain a better tracking performance. Extensive experiments show that the proposed DecoupleTNL achieves significantly better performance against the state-of-the-art TNL trackers. The prominent efficacy of DecoupleTNL would inspire researchers to pay attention to long short-term context modeling for tracking by the natural language specification.

Acknowledgements: This work was supported in part by the National Key Research and Development Program of China under Grant 2020AAA0106502, in part by the Nat-

ural Science Foundation of China under Grant 62073105, in part by the Natural Science Foundation of Heilongjiang Province of China under Grant ZD2022F002, in part by the State Key Laboratory of Robotics and System [Harbin Institute of Technology (HIT)] under Grant SKLRS-2019-KF-14 and Grant SKLRS-202003D, and in part by the Heilongjiang Touyan Innovation Team Program.

References

[1] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *Advances in Neural Information Processing Systems*, 33:9758–9770, 2020.

[2] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qiuhong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: A simplified architecture for visual object tracking. *arXiv preprint arXiv:2203.05328*, 2022.

[3] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8126–8135, 2021.

[4] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13608–13618, 2022.

[5] Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1769–1779, 2021.

[6] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5374–5383, 2019.

[7] Qi Feng, Vitaly Ablavsky, Qinxun Bai, Guorong Li, and Stan Sclaroff. Real-time visual object tracking with natural language description. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 700–709, 2020.

[8] Qi Feng, Vitaly Ablavsky, Qinxun Bai, and Stan Sclaroff. Robust visual object tracking with natural language region proposal network. *arXiv preprint arXiv:1912.02048*, 2019.

[9] Qi Feng, Vitaly Ablavsky, Qinxun Bai, and Stan Sclaroff. Siamese natural language tracker: Tracking by natural language descriptions with siamese trackers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5851–5860, 2021.

[10] Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. Aiatrack: Attention in attention for transformer visual tracking. In *European Conference on Computer Vision*, pages 146–164. Springer, 2022.

[11] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6269–6277, 2020.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Lianghua Huang, Yu Liu, Bin Wang, Pan Pan, Yinghui Xu, and Rong Jin. Self-supervised video representation learning by context and motion decoupling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13886–13895, 2021.

[15] Minji Kim, Seungkwan Lee, Jungseul Ok, Bohyung Han, and Minsu Cho. Towards sequence-level training for visual tracking. In *European Conference on Computer Vision*, pages 534–551. Springer, 2022.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[17] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *Advances in Neural Information Processing Systems*, 31, 2018.

[18] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[19] B. Li, J. Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. *CVPR*, pages 8971–8980, 2018.

[20] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees GM Snoek, and Arnold WM Smeulders. Tracking by natural language specification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6495–6503, 2017.

[21] 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. Springer, 2014.

[22] Ding Ma and Xiangqian Wu. Capsule-based object tracking with natural language specification. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1948–1956, 2021.

[23] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool. Transforming model prediction for tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8731–8740, 2022.

[24] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[25] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *CVPR*, 2017.

[26] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019.

[27] Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Transformer tracking with cyclic shifting window attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8791–8800, 2022.

[28] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019.

[29] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[30] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *European conference on computer vision*, pages 504–521. Springer, 2020.

[31] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1571–1580, 2021.

[32] Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13763–13773, 2021.

[33] Y. Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 37:1834–1848, 2015.

[34] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.

[35] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10448–10457, 2021.

[36] Bin Yu, Ming Tang, Linyu Zheng, Guibo Zhu, Jinqiao Wang, Hao Feng, Xuetao Feng, and Hanqing Lu. High-performance discriminative tracking with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9856–9865, 2021.