Progressive Semantic-Visual Alignment and Refinement for Vision-Language Tracking

Yanjie Liang, Qiangqiang Wu, Lin Cheng, Changqun Xia, Jia Li, *Senior Member, IEEE*

***Abstract*—In recent years, vision-language tracking has drawn emerging attention in the tracking field. The critical challenge for the task is to fuse semantic representations of language information and visual representations of vision information. For this purpose, several vision-language tracking methods perform early or late fusion to fuse visual and semantic features. However, these methods cannot take full advantage of the transformer architecture to excavate useful cross-modal context at various levels. To this end, we propose a new progressive joint vision-language transformer (PJVLT) to progressively align and refine visual embedding with semantic embedding for vision-language tracking. Specifically, to align visual signals with semantic signals, we propose to insert a semantic-aware instance encoder layer (SAIEL) into each intermediate layer of transformer encoder to perform progressive alignment of visual and semantic features. Furthermore, to highlight the multi-modal feature channels and patches corresponding to target objects, we propose a unified channel communication patch interaction layer (CCPIL), which is plugged into each intermediate layer of transformer encoder to progressively activate target-aware channels and patches of aligned multi-modal features for fine-grained tracking. In general, by progressively aligning and refining visual features with semantic features in the transformer encoder, our PJVLT can adaptively excavate well-aligned vision-language context at coarse-to-fine levels, therefore highlighting target objects at various levels for more discriminative tracking. Experiments on several tracking datasets show that the proposed PJVLT can achieve favorable performance in comparison with both conventional trackers and other vision-language trackers.**

***Index Terms*—vision-language tracking, progressive joint vision-language transformer, semantic-aware instance encoder, channel communication patch interaction.**

I. Introduction

OBJECT tracking is a fundamental task in computer vision with a wide range of applications, such as human computer interaction, video surveillance, autonomous driving, etc. The goal of this task is to continuously estimate the location (i.e., a bounding box) of an arbitrary object in a video sequence. Although many efforts have been made and various types of tracking methods have been developed over the past decades [1]–[13], there still exist various challenges in terms of tracking performance.

Most of the existing vision-only tracking frameworks build target models based on the visual appearance information given in the first frame. For instance, DCFNet [14] learns correlation filters by regressing the circular shifts of a sample from the first frame to the Gaussian-shaped labels. SiamFC [15] builds a Siamese model by matching the template at the first frame with the search region at the current frame. OSTrack [16] constructs a transformer model by performing bidirectional interaction between the template at the first frame and the search region at the current frame. However, when the target appearance continues to change over time, the target models may no longer adapt to the new target appearance.

To address the above issue, some researchers employ samples collected from the historical frames to refine the target model. For example, DiMP [17] employs multiple representative samples in the memory pool to predict a discriminative target model to improve its discriminativity. STMTrack [18] establishes a Siamese model by matching multiple templates stored in the memory pool with the search region at the current frame to improve the robustness. STSDL [19] collects various samples from historical frames to build a target model by performing joint spatio-temporal similarity and discrimination learning for accurate and robust tracking. However, these vision-only tracking methods model the target merely based on the visual information, and they are prone to drift in the case of severe appearance changes. Therefore, it is urgent to develop a new tracking framework to construct a target model by using rich semantic information rather than only with visual information.

In recent years, vision-language tracking has attracted emerging research interests and it provides a new human-machine interaction way for object tracking. Compared to tracking by bounding box specification, tracking by language specification has its natural advantages. On one hand, the bounding box can only provide static target appearance at the current frame, whereas the language description can identify a target with dynamic appearance across temporal frames. On the other hand, the bounding box cannot give specific semantics of the target, while the language description can specify exact semantics of the target (e.g., color, shape, class, etc.), which is beneficial to classify and localize the target. The earliest vision-language tracking methods [20]–[22] usually introduce language representations into tracking-by-detection

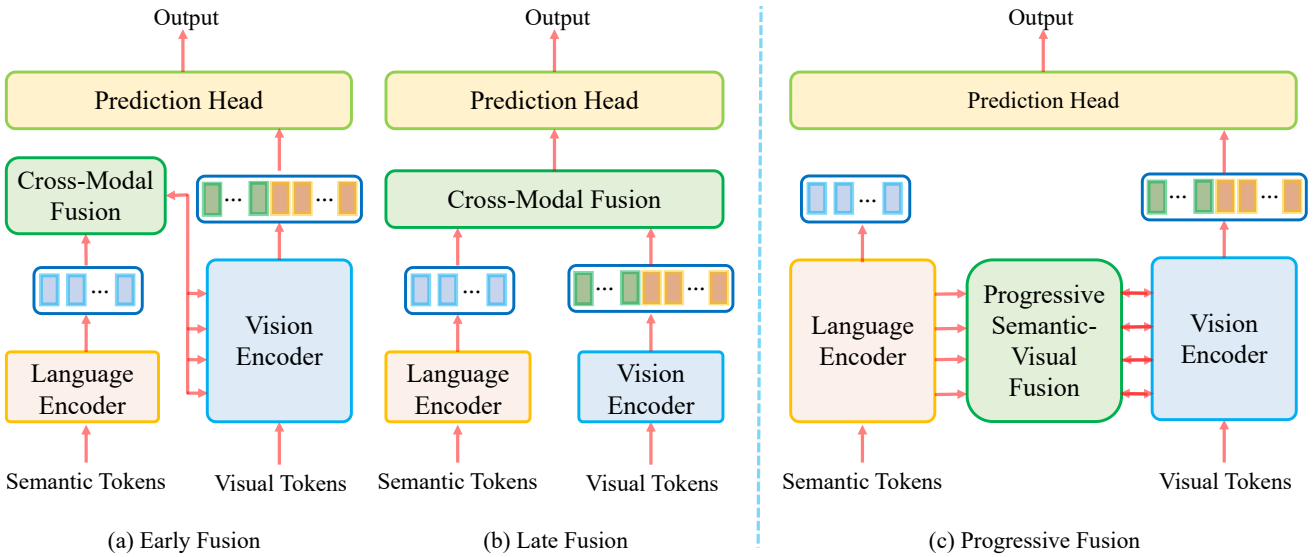
This work is supported by the National Natural Science Foundation of China (Grant No. 62132002, 62202249 and 62102206), the Major Key Project of PCL (Grant No. PCL2024A04-4), the Postdoctoral Science Foundation of China (Grant No. 2022M721732), the Postdoctoral Fellowship Program of China Postdoctoral Science Foundation (Grant No. GZC20233362), and the Chongqing Postdoctoral Innovative Talents Support Program (Grant No. CQBX202316).

Y. Liang, C. Xia are with Peng Cheng Laboratory, Shenzhen 518000, P.R.China (e-mail: liangyj@pcl.ac.cn, xiachq@pcl.ac.cn).

Q. Wu is with Department of Computer Science, City University of Hong Kong, Hong Kong 999077 (e-mail: qiangqwu2-c@my.cityu.edu.hk).

L. Cheng is with Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, Xiamen 361005, P.R.China (e-mail: lin_cheng163@163.com).

J. Li is with State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, P.R.China. (e-mail: jiali@buaa.edu.cn).



(a) Early Fusion (b) Late Fusion (c) Progressive Fusion

Fig. 1. Comparison of three different cross-modal fusion methods for vision-language tracking: (a) early fusion, (b) late fusion and (c) progressive fusion.

networks or Siamese networks to facilitate object tracking. However, due to the cross-modal feature misalignment, the performance of these methods is far behind the state-of-the-art. One of the critical challenges for vision-language tracking is performing cross-modal feature fusion between semantic words and visual pixels for target enhancement.

Existing vision-language tracking methods generally adopt early fusion [23] or late fusion [24]–[27] schemes for cross-modal feature fusion. As shown in Fig. 1(a), the tracking methods with early fusion scheme [23] firstly employ language encoders (i.e., transformers) to extract language features, and then the language features are interacted with the intermediate visual features from vision encoders (i.e., ConvNets or transformers) by exploiting cross-modal fusion modules to produce cross-modal representations. Despite exhibiting considerable potentials in tracking performance, the early fusion scheme cannot take advantage of the intermediate features of language encoders for cross-modal fusion at various semantic levels. As shown in Fig. 1(b), the tracking methods with late fusion scheme [24]–[27] firstly extracts visual and language features independently from the corresponding encoders, and then perform cross-modal fusion using vanilla transformer layers or transformer encoder-decoders to produce unified representations for target bounding box prediction. Although the late fusion scheme is able to gather vision and language modalities together, it does not conform to the human learning process, that is, integrating multiple sensors through various neurons before reasoning. Therefore, the potentiality of transformer for highlighting target is still far from being sufficiently excavated in the above paradigms, thus limiting the performance of vision-language tracking. To address the above issues, a possible solution is to develop a progressive joint vision-language transformer to simultaneously encode visual and semantic embedding.

In this paper, we propose a progressive joint vision-language transformer (PJVLT) for object tracking, where visual features from vision encoder and semantic features from language encoder are progressively interacted with each other, making visual context aware of its corresponding semantic context at various levels. The proposed PJVLT can fully exploit the multi-layer design of transformer to progressively align and refine visual embedding with semantic embedding at coarse-to-fine level. Fig. 1 compares our progressive fusion scheme with the early fusion scheme [23] and the late fusion scheme [24]–[27] for vision-language tracking. Specifically, to align visual features with semantic features, a semantic-aware instance encoder layer (SAIEL) is developed to align cross-modal signals at each intermediate layer of transformer encoder. Moreover, to enhance the patches and channels of multi-modal features corresponding to target objects, a channel communication patch interaction layer (CCPIL) is devised to activate target-aware patches and channels of multi-modal features flowing to the next layer of transformer encoder. Experimental results on several prevalent tracking datasets show that the proposed PJVLT performs favorably against other state-of-the-art tracking methods. The contributions of this paper can be summarized as the following four-folds:

- An end-to-end trainable progressive joint vision-language transformer (PJVLT) is proposed for vision-language tracking, which fully exploits multi-layer design in transformer encoder to fuse visual features and semantic features from coarse-to-fine levels in a progressive manner.
- An effective and efficient semantic-aware instance encoder layer (SAIEL) is designed to align visual patches with semantic sentences, which allows for deep excavation of vision-language context at intermediate layers of transformer encoder.
- A unified channel communication patch interaction layer (CCPIL) is devised to refine channels/patches of aligned visual-semantic features, which is capable of highlighting target-aware multi-modal feature channels/patches at intermediate layers of transformer encoder.
- Experiments are performed on four challenging tracking datasets to show that the proposed PJVLT can achieve the state-of-the-art performance with a real-time speed for vision-language tracking.

II. Related work

In this section, we briefly review two categories of tracking methods: vision-only tracking methods and vision-language tracking methods.

A. Vision-Only Tracking Methods

In the past decades, vision-only tracking has been developed rapidly to address the challenges of deformation, occlusion, background clutter, etc. The researchers in this community have developed various categories of vision-only tracking methods [28]–[30], such as tracking by deep classifiers [31]–[33], tracking by deep correlation filters [34]–[36], tracking by deep Siamese networks [37]–[39] and tracking by deep discriminant models [17], [40], [41].

In recent years, transformers have been widely applied in many computer vision tasks (e.g., video classification, object detection, visual tracking) and they have become a standard configuration to reach the state-of-the-art performance. The great success can be attributed to the attention layers in transformers that allow for deep feature interactions. Nowadays, tracking by transformers [16], [42]–[50] has become the most prevalent tracking methods, and these methods can be roughly categorized into the following three types.

The first type of transformer-based tracking methods [42], [43] typically use transformers to predict discriminative features for tracking. For instance, based on a transformer architecture, DTT [42] firstly feeds a reference frame into a transformer encoder, and then it feeds the encoded features into a transformer decoder to predict discriminative features of a test frame for target localization. TrDiMP [43] uses encoded features of reference frames to train a discriminative target model, which is further convolved with decoded features of test frame for discriminative tracking.

The second type of transformer-based tracking methods [44]–[47] stack features of both template and search region with transformers. For instance, TransT [44] uses multiple attention layers to fuse features for target classification and regression. Following the paradigm of DETR [51], STARK [45] adopts a full transformer to mix the features of template and search region for bounding box prediction. ToMP [46] also employs another full transformer from DETR [51] to predict the weights of a target classifier and a bounding box regressor. CSWinTT [47] elevates the attention from pixel-level to window-level by introducing multi-scale cyclic shifting window attention into a transformer for object tracking.

The third type of transformer-based tracking methods [16], [48]–[50] build one-stream unified tracking frameworks with transformers. MixFormer [48] employs a compact transformer to unify feature extraction and feature matching by designing iterative mixed attention modules for end-to-end tracking. OSTrack [16] constructs a neat transformer that combines feature learning and relation modeling by allowing for bidirectional information interaction between template and search region. To alleviate the target-background confusion, GRM [49] extends the relation model in OSTrack to a generalized relation model based on adaptive token division to improve the discriminability. CTTrack [50] introduces a correlative masked decoder into a one-stream framework to enhance the robustness of a compact transformer tracker. In [52], the authors employ the DropMAE model to replace the MAE model as a strong pre-trained backbone of one-stream framework to achieve better tracking performance. ARTrack [53] introduces spatio-temporal prompts into one-stream tracking framework for autoregressive tracking.

Despite achieving favorable tracking performance, most of the existing vision-only tracking methods only consider the vision information (i.e., the template image) while ignoring the language information (i.e., the natural language), which may be also useful for object tracking. In fact, language and vision are complementary cues. In contrast to the feature interactions between template and search region, our alternative is to explore the feature interactions between multi-modalities (i.e., vision and language) in the transformer paradigm. Therefore, we propose a progressive joint vision-language transformer (PJVLT), which takes advantage of both template image and natural language to facilitate vision-language tracking.

B. Vision-Language Tracking Methods

Natural language expressions are composed of high-level semantics and have been exploited to facilitate vision tasks, such as visual grounding [24], [54], [55], image segmentation [56], [57], video object segmentation [58], [59], video object tracking [22]–[24]. These vision-language models typically integrate a language model and a vision model to foster a common embedding space for both language and vision. The recent vision-language models firstly extract vision features and language features using Siamese networks, and then perform depth-wise convolution between language features and visual features.

For video object tracking, the work [20] initially defines three versions of tracking by language specification and validates that tracking by vision-language achieves the best performance among the three versions. After that, the work [21] derives a deep tracking-by-detection formulation that can take advantage of natural language expressions for vision-language tracking; the work [22] introduces natural language information into Siamese paradigm by carefully designing Siamese natural language region proposal networks to perform vision-language tracking. However, these vision-language trackers treat vision and language as independent cues until the final fusion stage, and the performance of these vision-language tracking methods is still far behind the state-of-the-art.

The state-of-the-art vision-language tracking methods with transformer paradigms either perform early fusion [23] or late fusion [24]–[27] to aggregate vision and language modalities together. For early fusion, VLT$_{\text{SiamCAR}}$ [23] and VLT$_{\text{TransT}}$ [23] align vision modality and language modality by embedding modality mixture modules into a convolutional network and a transformer network to learn unified vision-language representations, which shows great potentials of vision-language tracking to achieve the state-of-the-art performance. However, they are unable to leverage the multi-layer design of language encoders to perform progressive feature alignment at various levels. For late fusion, JointNLT [24] and

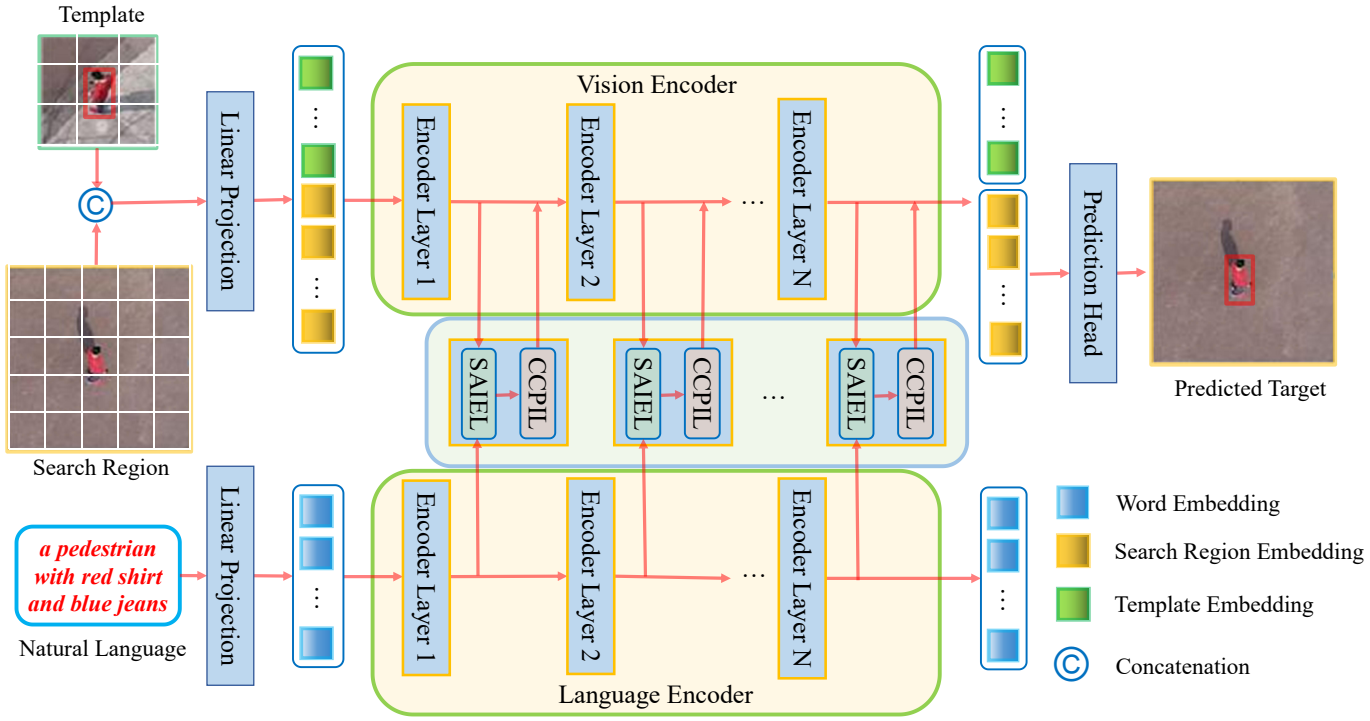


Fig. 2. Overall framework of our PJVLT model. The inputs of our PJVLT model consist of a template, a search region and a natural language expression. The template and the search region are patchfied, concatenated and projected to produce input visual tokens, and the language expression is projected by using a language encoder to generate input semantic tokens. Afterwards, the input visual tokens and semantic tokens are interacted with each other within our progressive joint vision-language transformer encoder. Finally, the output visual tokens corresponding to the search region are reshaped into feature maps to predict the target bounding box by using a prediction head.

MMTrack [25] perform multi-source multi-modal interactions within vanilla transformer layers for vision-language tracking. QueryNLT [26] and TransNLT [27] employ transformer encoder-decoders to integrate language and visual modalities for context-aware vision-language tracking. In spite of achieving favorable performance, they fail to effectively exploit the intermediate layers of vision/language encoder to excavate the vision-language context. In contrast to the modality mixture modules for early fusion and the transformer layers/encoder-decoders for late fusion, we propose a progressive joint vision-language transformer (PJVLT) to gradually align and refine visual patches with semantic words at multiple levels (e.g., pixel, semantic and class).

III. Methodology

In this section, we first give a brief overview of our vision-language tracking method in Sec. III-A. Subsequently, we introduce our progressive joint vision-language encoding scheme in Sec. III-B. Then, we propose our semantic-aware instance encoder and channel communication patch interaction in Sec. III-C and Sec. III-D, respectively. Finally, we describe the prediction head and the training loss of our network in Sec. III-E.

A. Framework Overview

The overall pipeline of our progressive joint vision-language transformer (PJVLT) model is illustrated in Fig. 2. It is designed to fully exploit the layer-wise structure of transformer to perform progressive visual-semantic alignment and refinement in joint vision-language transformer encoder, so that visual features from each intermediate layer of vision transformer encoder can be aware of the corresponding semantic context from each intermediate layer of language transformer encoder. To our best knowledge, the proposed PJVLT model is the first work to adopt a novel progressive multi-modal encoding method for vision-language tracking.

As shown in Fig. 2, the proposed PJVLT model mainly consists of linear projection layers, a vision transformer encoder, a language transformer encoder, semantic-aware instance encoder layers (SAIELs), channel communication patch interaction layers (CCPILs), and a target prediction head. The language encoder takes a natural language expression as input, and it is responsible for extracting semantic tokens. The vision encoder takes a template image and a search region image as inputs, and it focuses on extracting visual tokens. Afterwards, SAIELs and CCPILs are responsible for progressively aggregating semantic tokens and visual tokens. Finally, the search region tokens are fed into the target prediction head for target classification and bounding box regression. The proposed PJVLT model can progressively align visual tokens with semantic tokens and refine aligned multi-modal tokens at coarse-to-fine levels during the encoding stage.

B. Progressive Joint Vision-Language Encoding

Given an input triple of a template, a search region and a natural language expression that identifies a target to be

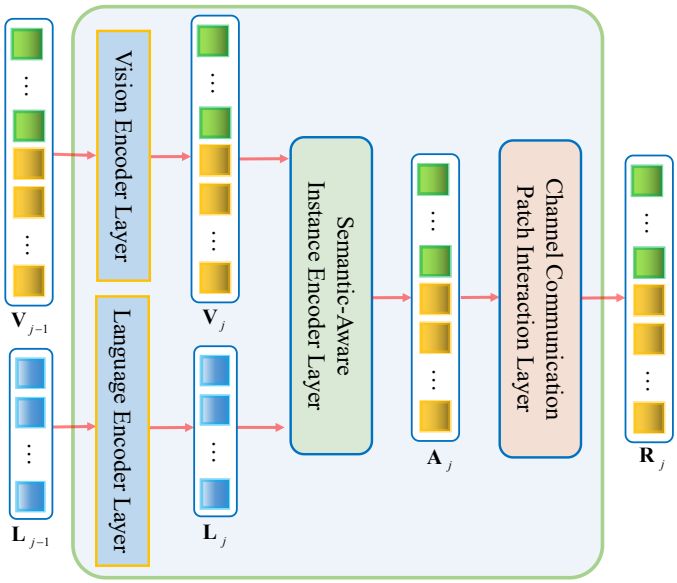


Fig. 3. Pipeline of our joint vision-language encoder layer. Each encoder layer takes visual tokens and semantic tokens from the previous layer as inputs, and it outputs multi-modal tokens. The visual/semantic tokens from the $(j-1)$-th layer $\mathbf{V}_{j-1}$/$\mathbf{L}_{j-1}$ are taken as the inputs of vision/language encoder layer to produce a set of visual/semantic tokens at the j-th layer $\mathbf{V}_j$/$\mathbf{L}_j$. Then, the visual tokens $\mathbf{V}_j$ and semantic tokens $\mathbf{L}_j$ are fed into a semantic-aware instance encoder layer to produce a set of aligned multi-modal tokens at the j-th layer $\mathbf{A}_j$. Finally, the aligned semantic-visual tokens $\mathbf{A}_j$ are fed into a channel communication patch interaction layer to output a set of refined multi-modal tokens at the j-th layer $\mathbf{R}_j$.

tracked, the proposed model outputs a bounding box of the target in the search region.

To convert the natural language expression into high-dimensional word embedding, we employ a pre-trained language encoder to progressively extract semantic tokens. The semantic tokens from j-th intermediate layer are denoted as $\mathbf{L}_j \in \mathbb{R}^{C\times T}$, where C and T are the number of channels and words, respectively. To convert the template and the search region into visual embedding, we firstly slice and concatenate them, and then use a pre-trained vision encoder to progressively extract visual tokens. The visual tokens from j-th intermediate layer of vision encoder are referred as $\mathbf{V}_j \in \mathbb{R}^{C\times N}$, where C and N are the number of channels and patches, respectively.

After extracting semantic tokens from the pre-trained language encoder and visual tokens from the pre-trained vision encoder in parallel, we merge visual tokens $\mathbf{V}_j$ and semantic tokens $\mathbf{L}_j$ at each intermediate layer of our progressive joint vision-language encoder. Each joint vision-language encoder layer consists of a vision encoder layer Θ_j , a language encoder layer Π_j, a semantic-aware instance encoder layer Φ_j, and a channel communication patch interaction layer Ψ_j.

As depicted in Fig. 3, at the j-th layer, the model produces and fuses visual and semantic tokens by three steps as follows. Firstly, the visual tokens from the $(j-1)$-th layer (denoted as $\mathbf{V}_{j-1}$) are taken as the input of the vision encoder layer Θ_j to generate a set of visual tokens at the j-th layer, which are referred as $\mathbf{V}_j \in \mathbb{R}^{C\times N}$. Meanwhile, the semantic tokens from the $(j-1)$-th layer (denoted as $\mathbf{L}_{j-1}$) are fed into the language encoder layer Π_j to produce a set of semantic tokens at the j-th layer, which are denoted as $\mathbf{L}_j \in \mathbb{R}^{C\times T}$. Then, the visual tokens $\mathbf{V}_j$ and the semantic tokens $\mathbf{L}_j$ are fed into the SAIEL Φ_j to generate a set of aligned multi-modal tokens at j-th layer, which are referred as $\mathbf{A}_j \in \mathbb{R}^{C\times N}$. Finally, the aligned semantic-visual tokens $\mathbf{A}_j$ are taken as the input of CCPIL Ψ_j to produce a set of refined multi-modal tokens at the j-th layer, which are referred as $\mathbf{R}_j \in \mathbb{R}^{C\times N}$.

The vision encoder layer is carried from the encoder layer in ViT [60], and the language encoder layer is inherited from the encoder layer in BERT [61]. The proposed SAIEL is carefully designed to densely align visual clues in visual tokens with semantic context in semantic tokens, which is described in Sec. III-C. The proposed CCPIL is specially devised to activate target-aware channels and patches of aligned visual-semantic tokens for fine-grained object tracking, which is described in Sec. III-D.

C. Semantic-Aware Instance Encoder

To effectively discriminate a target from its surrounding background and accurately estimate its bounding box, it is important to progressively align visual tokens with semantic tokens of the target across vision-language modalities. A possible solution is to fuse the representation of every visual patch with the representation of a language sentence, and perform joint vision-language representation learning to obtain discriminative features to distinguish a target from its background. In comparison with the previous method [24] that firstly extracts visual/semantic tokens independently from vision/language encoders and then performs visual-semantic fusion. The proposed semantic-aware instance encoder layer (SAIEL) allows progressive interactions between patches in visual tokens and sentence in semantic tokens, which is effective to fully exploit the transformer encoder layers to excavate the vision-language context.

Fig. 4 schematically depicts the proposed semantic-aware instance encoder layer (SAIEL). Taken visual tokens $\mathbf{V}_j \in \mathbb{R}^{C\times N}$ from the j-th vision encoder layer and semantic tokens $\mathbf{L}_j \in \mathbb{R}^{C\times T}$ from the j-th language encoder layer as inputs, SAIEL is proposed to perform patch-word alignment. For every visual patch, SAIEL aggregates the semantic tokens across channel dimension to produce position-specific, word-level feature maps, which encode the semantic word information around the local neighborhood visual patches. Firstly, SAIEL projects the visual tokens $\mathbf{V}_j$ and semantic tokens $\mathbf{L}_j$ into a latent space as follows:

$$\mathbf{V}_q = \theta_q(\mathbf{V}_j), \mathbf{L}_k = \theta_k(\mathbf{L}_j), \mathbf{L}_v = \theta_v(\mathbf{L}_j), \tag{1}$$

where θ_q is the query projection function for $\mathbf{V}_j$. θ_k and θ_v are the key and value projection functions for $\mathbf{L}_j$, respectively. Then, SAIEL performs Gumbel-Softmax [62] operation on $\mathbf{L}_v$ to choose a set of words in the sentence as follows:

$$\hat{\mathbf{L}}_v = Gumbel - Softmax(\mathbf{L}_v) \odot \mathbf{L}_v, \tag{2}$$

where $\odot$ denotes the word-wise multiplication. Afterwards, SAIEL performs cross-attention to produce a set of attentional

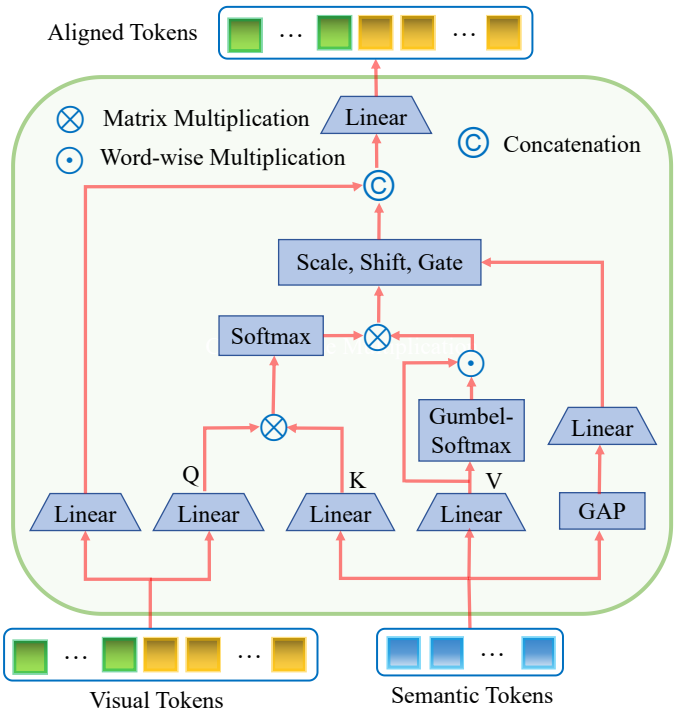


Fig. 4. Pipeline of our semantic-aware instance encoder layer (SAIEL). SAIEL is specifically designed to produce a set of aligned visual-semantic tokens. SAIEL firstly performs cross-attention to produce attentional semantic tokens, where the projected visual tokens are taken as queries and the projected semantic tokens are considered as keys and values. Then, it projects the pooled semantic tokens to estimate the scale, shift and gate parameters to modulate the attentional semantic tokens as spatial-aware semantic tokens. Afterwards, it concatenates the produced spatial-aware semantic tokens and the projected visual tokens. Finally, it projects the concatenated tokens to produce the aligned visual-semantic tokens.

semantic tokens, which are denoted as $\mathbf{L}_{attn} \in \mathbb{R}^{C\times N}$ as follows:

$$\mathbf{L}_{attn} = Softmax(\frac{\mathbf{V}_q^T \mathbf{L}_k}{\sqrt{C}})\hat{\mathbf{L}}_v. \tag{3}$$

In parallel, SAIEL performs global average pooling on the semantic tokens $\mathbf{L}_j$ across word dimension to produce a global semantic token $\mathbf{L}_j^g$, which is further projected to estimate the scale, shift and gate parameters α, β, γ as follows:

$$\alpha, \beta, \gamma = \theta_w(\mathbf{L}_j^g), \tag{4}$$

where θ_w is the linear projection function for $\mathbf{L}_j^g$. Finally, SAIEL produces the spatial-aware semantic tokens $\mathbf{L}_o$ as follows:

$$\mathbf{L}_o = \gamma((1+\alpha)\mathbf{L}_{attn} + \beta). \tag{5}$$

In implementation, we adopt 1×1 convolution to implement the key and value projection functions θ_k and θ_v; we employ 1×1 convolution and instance normalization to implement the query projection functions θ_q. In Eq. (3), we perform the scaled dot-product attention, where the visual embedding is taken as the query and the semantic embedding is considered as the key and value.

After attaining the spatial-aware semantic tokens $\mathbf{L}_o$, which keeps the same spatial size as the visual tokens $\mathbf{V}_j$, we perform feature integration to produce the aligned visual-semantic tokens $\mathbf{A}_j$ by concatenation across channels. Specifically, $\mathbf{A}_j$ is generated as follows:

$$\mathbf{V}_o = \theta_m(\mathbf{V}_j), \tag{6}$$

$$\mathbf{A}_j = \theta_o(\mathbf{V}_o \copyright \mathbf{L}_o), \tag{7}$$

where © denotes the concatenation operation, θ_m is the visual projection function for $\mathbf{V}_j$, and θ_o is the final visual-semantic projection function for the concatenated features. In implementation, we employ 1×1 convolution followed by a GELU activation function to implement θ_m and θ_o.

As shown in Fig. 4, through the semantic-aware instance encoder layer, the semantic tokens and visual tokens are firstly aligned at the spatial level to produce the spatial-aware semantic tokens. Afterwards, the spatial-aware semantic tokens and visual tokens are concatenated and projected to produce the aligned multi-modal tokens. The proposed SAIEL has a capability to promote aligning visual patches and semantic sentences for cross-modal representation learning.

In contrast to existing cross-attention layers, the proposed SAIEL is a plug-and-play layer in our PJVLT model and it has four-folds specific characteristics as follows: (1) *Better Semantic-to-Visual Transfer:* Inspired by the image diffusion network [63], we zero-initialize the linear projection layer θ_w to regress the scale, shift and gate parameters, which facilitates our SAIEL gradually transfer the knowledge from global semantic token to visual patches. (2) *Gumbel-Softmax-based Word Selection:* As a Gumbel-Softmax distribution can be smoothly annealed into a categorical distribution [62], we adopt a unique design that applies the Gumbel-Softmax operation on the value $\mathbf{L}_v$ to estimate the importance score of each word in a sentence, facilitating the selection of a word set for better cross-attention. (3) *Higher Efficiency:* We implement the multi-head cross-attention block using FlashAttention [64], which is both time-efficient and memory-efficient when we perform computation between the dense visual patches and the semantic words. (4) *Computation in Compressed Latent Space:* We project the visual/semantic features from an original space to a compressed latent space for efficient computation, and then re-project the visual-semantic features from the compressed latent space to the original space for dimension restoration. Thanks to our effective and efficient SAIEL, the semantic information can be injected into the visual features using the same SAIEL in a progressive manner.

D. Channel Communication Patch Interaction

As described in Sec. III-B, the semantic-aware instance encoder layer (SAIEL) is responsible for progressive aligning visual tokens with semantic tokens. However, different feature channels of aligned features usually corresponds to various semantics [65]. For instance, some feature channels represent the foreground object "dog", while other channels may encode the background distractor "cat". Thus, it is preferable to highlight the aligned feature channels corresponding to target objects and eliminate the aligned feature channels representing background distractors. Moreover, the detailed information between different patches is essential for fine-grained object recognition [66]. For instance, the full interaction of the

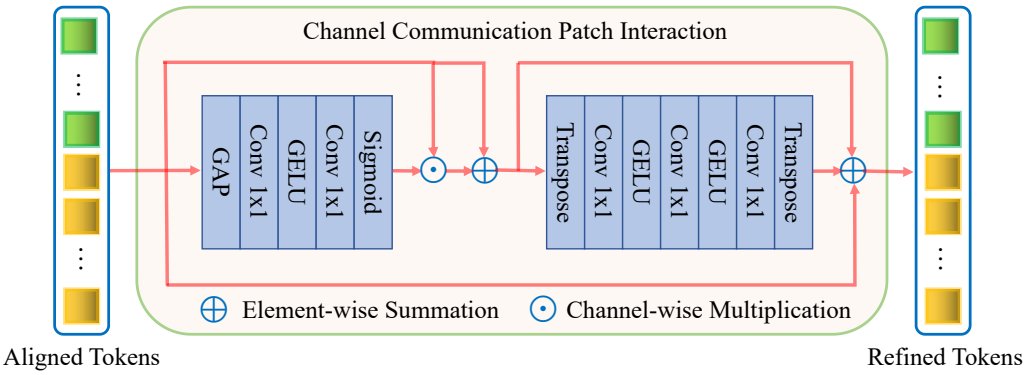


Fig. 5. Pipeline of the proposed channel communication patch interaction layer (CCPIL). CCPIL is devised to refine channels and patches of aligned multi-modal tokens for target-aware tracking. CCPIL firstly communicates the channels of aligned tokens by estimating the channel-wise weights to re-scale themselves. Afterwards, it projects the communicated tokens into a high dimension space for fine-grained patch interaction to estimate a residual of themselves to produce interacted tokens.

patches containing "head" of "dog" and the patches covering "body" of "dog" is beneficial to recognize the target object "dog". Therefore, it is desirable to refine the association between aligned visual-semantic feature patches. According to the above observation, we propose to insert a unified channel communication patch interaction layer (CCPIL) into each intermediate layer of transformer encoder to refine channels and patches of aligned visual-semantic tokens for target-aware multi-modal tracking.

Fig. 5 schematically illustrates the pipeline of the proposed CCPIL, which consists of a channel communication module (CCM) and a patch interaction module (PIM). CCM employs aligned multi-modal tokens $\mathbf{A}_j$ to predict a set of channel-wise weights to re-scale the set of multi-modal feature maps in $\mathbf{A}_j$. As different channels of multi-modal tokens are interdependent, CCM performs channel communication to compact the relevant channels and scatter the irrelevant channels by using a group compact attention $f_{gc}(\cdot)$, which can be mathematically formulated as follows:

$$f_{gc}(\mathbf{A}_j) = Sig(\sigma(\hbar(\mathbf{A}_j) \cdot \mathbf{W}_{p1}) \cdot \mathbf{W}_{p2}), \tag{8}$$

$$\mathbf{C}_j = f_{gc}(\mathbf{A}_j) \odot \mathbf{A}_j + \mathbf{A}_j, \tag{9}$$

where $\odot$ is the channel-wise multiplication operation, $\hbar(\cdot)$ denotes a global average pooling function, $\sigma(\cdot)$ represents a GELU function, and $Sig(\cdot)$ is a sigmoid activation function. $\mathbf{W}_{p1} \in \mathbb{R}^{C \times \frac{C}{N_g}}$ and $\mathbf{W}_{p2} \in \mathbb{R}^{\frac{C}{N_g} \times C}$ denote the parameters of two 1×1 convolution layers, respectively. N_g represents the number of channel groups, which is set to 4 in this paper.

Inspired by the concept of the manifold of interest in MobileNetV2 [66], PIM conducts patch interaction by using an inverted residual and a linear bottleneck to boost the representation power of communicated multi-modal tokens $\mathbf{C}_j$, which can be mathematically formulated as follows:

$$\begin{aligned} \mathbf{C}_e &= f_e(\mathbf{C}_j) = \sigma((\mathbf{C}_j)^T \cdot \mathbf{W}_e), \\ \mathbf{C}_s &= f_s(\mathbf{C}_e) = \sigma(\mathbf{C}_e \cdot \mathbf{W}_s), \\ \mathbf{C}_n &= f_n(\mathbf{C}_s) = \mathbf{C}_s \cdot \mathbf{W}_n, \end{aligned} \tag{10}$$

where $f_e(\cdot)$ is an expansion function, which is implemented by a 1×1 convolution layer with parameters $\mathbf{W}_e \in \mathbb{R}^{N_l \times N_h}$ followed by a GELU activation function. By using the expansion function, the length of tokens is expanded from a low dimension N to a high dimension N_h. $f_s(\cdot)$ is a selection function, which consists of a 1×1 convolution layer with parameters $\mathbf{W}_s \in \mathbb{R}^{N_h \times N_h}$ followed by a GELU activation function, to perform patch interaction. $f_n(\cdot)$ is a narrow linear bottleneck, which is implemented by a 1×1 convolution layer with parameters $\mathbf{W}_n \in \mathbb{R}^{N_h \times N}$, to project the interacted multi-modal tokens from the high dimension N_h to the original low dimension N. N_h is the number of patches in the high dimension space, which is set to 640 in this paper. Finally, PIM employs a shortcut to produce the refined multi-modal tokens $\mathbf{R}_j$ as follows:

$$\mathbf{V}_{j+1} = \mathbf{R}_j = \mathbf{C}_n + \mathbf{C}_j + \mathbf{A}_j. \tag{11}$$

As shown in Fig. 5, with channel communication, the aligned multi-modal tokens are global average pooled along the token dimension to estimate each channel importance to reweigh the aligned tokens. Afterwards, through patch interaction, the communicated multi-modal tokens are further expanded to the high dimension space for patch interaction to produce the refined multi-modal tokens. The proposed CCPIL has capability to promote activating some target-aware channels and patches for cross-modal representation learning.

In contrast to existing attention layers, our CCPIL has its special characteristics as follows: (1) CCPIL introduces CCM to allow for effective communication between semantic channels, which is beneficial to gather intra-group semantic channels and separate inter-group semantic channels. (2) CCPIL employs PIM to perform fine-grained patch interaction between patch tokens, which is effective to separate target patches from background patches. Thanks to the unified CCPIL, target-aware channels and patches of aligned multi-modal features can be progressively enhanced in our PJVLT model.

E. Prediction Head and Training Loss

For bounding box prediction, we first convert the search region tokens from the output of progressive joint vision-language transformer encoder into a set of feature maps and

then feed them into a fully convolutional network (FCN) for prediction. FCN consists of M stacked Conv-BN-ReLU layers, and it outputs three branches to predict a classification score map $\mathbf{P} \in [0,1)^{H\times W}$, a regression offset map $\mathbf{O} \in [0,1)^{2\times H\times W}$ and a regression size map $\mathbf{S} \in [0,1)^{2\times H\times W}$, where H and W denotes the height and width of these maps. The position with the highest score in the score map is considered as the target center, i.e., $(x_p, y_p) = \arg\max_{(x,y)} \mathbf{P}_{xy}$, and the corresponding regressed coordinates in the offset map and the size map are employed to predict the target bounding box as:

$$(x_t, y_t) = (x_p, y_p) + (\mathbf{O}(0, x_p, y_p), \mathbf{O}(1, x_p, y_p)), \tag{12}$$

$$(h_t, w_t) = (\mathbf{S}(0, x_p, y_p), \mathbf{S}(1, x_p, y_p)). \tag{13}$$

In the training stage, a focal loss [67] weighted by a Gaussian kernel is employed for target classification; a L_1 Loss [16] and a GIoU loss [68] is used for bounding box regression. For target classification, given a ground-truth target center, we calculate its corresponding low-resolution equivalent $\hat{\mathbf{p}} = (\hat{p}_x, \hat{p}_y)$, and then produce a ground-truth classification score map by using a Gaussian kernel as:

$$\hat{\mathbf{P}}_{xy} = \exp(-\frac{(x-\hat{p}_x)^2 + (y-\hat{p}_y)^2}{2\sigma_p^2}), \tag{14}$$

where σ_p^2 is a size-adaptive standard deviation. The focal loss weighted by a Gaussian kernel is mathematically formulated as:

$$\mathcal{L}_{cls} = -\sum_{xy} \begin{cases} (1-\mathbf{P}_{xy})^a \log(\mathbf{P}_{xy}), \text{if } \hat{\mathbf{P}}_{xy} = 1 \\ (1-\hat{\mathbf{P}}_{xy})^b (\mathbf{P}_{xy})^a \log(1-\mathbf{P}_{xy}), \text{others} \end{cases} \tag{15}$$

where a and b are parameters of the focal loss, which are set to 2 and 4, respectively. For bounding box regression, given the ground-truth and predicted target bounding boxes $\hat{B}$ and B. The GIoU loss is defined as:

$$\mathcal{L}_{iou} = 1 - \frac{|B \cap \hat{B}|}{|B \cup \hat{B}|} + \frac{|C - (B \cup \hat{B})|}{|C|}, \tag{16}$$

where C denotes the smallest enclosing convex bounding box for $\hat{B}$ and B.

The total loss of the prediction head is a combination of a focal loss for target center classification, a GIoU loss and a L_1 loss for bounding box coordinate regression as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \mu_{iou}\mathcal{L}_{iou} + \mu_{L_1}\mathcal{L}_1, \tag{17}$$

where μ_{iou} and μ_{L_1} are regulation parameters, which are set to 2 and 5, respectively.

IV. Experiments

In this section, we first give implementation details in Sec. IV-A. Then, we provide comparison results on four prevalent tracking datasets in Sec. IV-B. Next, we perform ablation studies to validate the effectiveness of our PJVLT model in Sec. IV-C. Finally, we show some examples for qualitative comparison.

A. Implementation Details

Network Structure. The network of our PJVLT model consists of a base ViT [60] backbone, a base BERT [61] backbone, SAIELs, CCPILs and a box prediction head. The ViT backbone with 12 layers is employed to encode visual features for both template and search region, and it is initialized with the pre-trained backbone of MAE [69]. The base BERT backbone with 12 layers is employed to encode semantic features, and it is initialized with the official pre-trained backbone. The box prediction head with 4 stacked Conv-BN-ReLU layers is employed to predict three outputs (i.e., a classification score map, an offset score map and a size score map), and it is initialized with random weights. The proposed SAIELs and CCPILs, which are employed for semantic-visual feature alignment and multi-modal feature refinement, are also randomly initialized.

Off-line Training. We employs frame pairs and natural language expressions from the training splits of LaSOT [70], GOT10K [71], TrackingNet [72], and COCO [73] to train our PJVLT model. For the GOT10K dataset, the object class information, the motion class information, the major class information and the root class information of a video are concatenated as its natural language expression. For the TrackingNet dataset, the object class information of a video is taken as its natural language expression. For the COCO dataset, the category information and the supercategory information of an object in a still image are concatenated as the natural language expression of the image.

For the video datasets, we choose a pair of video frames and a natural language expression annotated at the first video frame as a training sample. For the image datasets, we choose duplicated still images and a natural language expression annotated at the still image as a training sample. One frame/image is chosen as the template, the other frame/image is treated as the search region, and the natural language expression is the input of language encoder. We directly sample the training samples from the same video sequence or the still image, and we apply brightness jitter and horizontal flip for data augmentation.

We employ the AdamW optimizer with a weight decay of 1.0×10^{-4} to train our model for 300 epochs with a total batch size of 128, where the initial learning rate for the progressive joint vision-language transformer encoder (including the ViT backbone, the BERT backbone, SAIELs and CCPILs) is set to 1.0×10^{-5}, and the initial learning rate for the prediction head is set to 1.0×10^{-4}. The learning rate is adjusted by a decrease factor of 0.1 after 240 epochs. For the model trained on the GOT10K training set, we train our model for 100 epochs, where the learning rate is adjusted after 60 epochs.

Online Inference. In the tracking procedure, we feed the template at the first video frame, the search region at the current video frame and the language expression for the video sequence into the well-trained network for bounding box prediction. we also employ a Hanning window to penalize the large movement across consecutive video frames as the common practice [15], [37], [38]. To be more specific, the predicted classification score map is multiplied by the Hanning window of the same spatial size as the final classification score

TABLE I
COMPARISON RESULTS ON THE FOUR PREVALENT SINGLE OBJECT TRACKING DATASETS (LASOT [70], LASOTEXT [74], GOT10K [71] AND TNL2K [75]). THE FIRST, SECOND AND THIRD BEST RESULTS ARE HIGHLIGHTED BY **RED**, **BLUE** AND **GREEN**, RESPECTIVELY.

Method	Publication	LaSOT			LaSOText			GOT10K			TNL2K			Speed
		AUC	NP	P	AUC	NP	P	AO	$SR_{0.50}$	$SR_{0.75}$	AUC	NP	P	
SiamRCNN [76]	CVPR20	64.8	72.2	-	-	-	-	64.9	72.8	59.7	52.3	-	52.8	5
LTMU [77]	CVPR20	57.2	-	57.2	41.4	**49.9**	47.3	-	-	-	-	-	-	13
AutoMatch [78]	ICCV21	58.3	-	59.9	37.6	-	43.0	65.2	76.6	54.3	47.2	-	43.5	50
SiamCAR [79]	CVPR20	50.7	-	51.0	33.9	-	41.0	56.9	67.0	41.5	35.3	43.6	38.4	52
TransT [44]	CVPR21	64.9	73.8	69.0	44.8	-	52.5	67.1	76.8	60.9	50.7	**57.1**	51.7	32
TrDiMP [43]	CVPR21	63.9	-	66.3	-	-	-	67.1	76.8	60.9	-	-	-	26
STARK [45]	ICCV21	67.1	77.0	71.2	**47.7**	-	**54.9**	68.0	77.7	62.3	-	-	-	32
CSWinTT [47]	CVPR22	66.2	75.2	70.9	-	-	-	69.4	78.9	65.4	-	-	-	12
UTT [80]	CVPR22	64.6	-	67.2	-	-	-	67.2	76.3	60.5	-	-	-	25
ToMP [46]	CVPR22	67.6	78.0	72.2	45.9	-	-	-	-	-	-	-	-	20
MixFormer [48]	CVPR22	**69.2**	**78.7**	**74.7**	-	-	-	70.7	80.0	67.8	-	-	-	25
OSTrack [16]	ECCV22	68.7	78.1	**74.6**	47.4	**57.3**	53.3	71.0	80.4	68.2	54.3	-	-	**93**
SimTrack [81]	ECCV22	**69.3**	**78.5**	74.0	-	-	-	68.6	78.9	62.4	**54.8**	-	**53.8**	40
SwinTrack [82]	NeurIPS22	67.2	-	70.8	47.6	-	53.8	**71.3**	**81.9**	64.5	53.0	-	53.2	**98**
CTTrack [50]	AAAI23	67.8	77.8	74.0	-	-	-	**71.3**	80.7	**70.3**	-	-	-	40
SNLT [22]	CVPR21	54.0	-	57.6	26.2	-	30.0	43.3	50.6	22.1	27.6	-	41.9	50
TNL2K-II [75]	CVPR21	51.0	-	55.0	-	-	-	-	-	-	42.0	**50.0**	42.0	-
$VLT_{SiamCAR}$ [23]	NeurIPS22	65.2	-	69.1	44.7	-	51.6	61.4	72.4	52.3	49.8	-	51.0	43
VLT_{TransT} [23]	NeurIPS22	67.3	-	72.1	**48.4**	-	**55.9**	69.4	81.1	64.5	53.1	-	53.3	35
JointNLT [24]	CVPR23	60.4	-	63.6	-	-	-	-	-	-	**56.9**	-	**58.1**	39
PJVLT*	Ours	65.8	75.3	71.0	-	-	-	**72.8**	**83.0**	**70.4**	-	-	-	**54**
PJVLT	Ours	**69.0**	**78.4**	**74.8**	**48.5**	**58.8**	**55.0**	**76.2**	**85.8**	**74.2**	**56.2**	**72.9**	**57.2**	**54**

map, and the position corresponding to the highest score in the final classification score map is considered as the target center. The regressed coordinates corresponding to the target center in the offset score map and the size score map are jointly used to determine the target bounding box.

In experiments, the size of the template and the search region is set to 128×128 and 256×256 pixels, respectively. The length of the natural language expression is set to 20. The proposed PJVLT model is trained on 2 Tesla T4 GPUs, and it is tested on a single GPU with a tracking speed of 54 FPS.

B. Comparison with State-of-the-Arts

In this section, We compare the proposed PJVLT with fifteen vision-only tracking methods (including SiamRCNN [76], LTMU [77], AutoMatch [78], SiamCAR [79], TransT [44], TrDiMP [43], STARK [45], CSWinTT [47], UTT [80], ToMP [46], MixFormer [48], OSTrack [16], SimTrack [81], SwinTrack [82] and CTTrack [50]) and five prevalent vision-language tracking methods (including SNLT [22], TNL2K-II [75], $VLT_{SiamCAR}$ [23], VLT_{TransT} [23] and JointNLT [24]) on four challenging datasets.

LaSOT [70]. The LaSOT test set consists of 280 videos, where the average video length is over 2500 frames. The LaSOT test set adopts distance precision (P), normalized distance precision (NP) and area under curve (AUC) as the evaluation metrics. Tab. I reports the comparison results of PJVLT and the other state-of-the-art methods. As illustrated in Tab. I, we observe that our PJVLT with a joint vision-language encoder achieves the competitive performance with an AUC score of 69.0%, a NP score of 78.4% and a P score of 74.8%. When only trained on the LaSOT train set, our model achieves an AUC score of 65.8%, a NP score of 75.3% and a P score of 71.0%. In comparison with the vision-only tracking methods (e.g., TransT, STARK, CSWinTT, UTT, ToMP, MixFormer, OSTrack, SimTrack, SwinTrack, CTTrack), the proposed PJVLT achieves the competitive performance on the LaSOT test set. The reason why the proposed PJVLT cannot achieve significant performance gains in comparison with the state-of-the-art vision-only tracking methods (i.e., MixFormer and SimTrack) is that the language descriptions specified at the first frame of video sequences are not sufficient to support the long-term tracking for the dynamic target objects. Nevertheless, the training time of PJVLT (i.e., 300 epochs) is much less than the training time of MixFormer (i.e., 500 epochs) and SimTrack (i.e., 500 epochs), and our PJVLT (i.e., 54 fps) runs faster than MixFormer (i.e., 25 fps) and SimTrack (i.e., 40 fps). In comparison with vision-language tracking methods (i.e., SNLT, TNL2K-II, $VLT_{SiamCAR}$ VLT_{TransT} and JointNLT), our PJVLT sets a new state-of-the-art, surpassing the second best vision-language tracking method VLT_{TransT} by

1.7/2.7 absolute points in terms of the AUC/P score.

LaSOText [74]. LaSOText is an extension of LaSOT, which consists of 150 extra videos. The video sequences in LaSOText are challenging as many unseen objects are distracted by similar distractors. As reported in Tab. I, our PJVLT achieves favorable performance with an AUC score of 48.5%, a NP score of 58.8% and a P score of 55.0%. In comparison with the vision-only tracking methods, our PJVLT achieves the best performance, and the training time of PJVLT (i.e., 300 epochs) is nearly half of the training time of the best vision-only tracking method STARK (i.e., i.e., 500 epochs for localization in the first stage and 50 epochs for classification in the second stage). Note that both our PJVLT and STARK use 6.0×10^4 triplets per epoch to train their models. Compared with the five vision-language tracking methods, our PJVLT obtains the best AUC/NP score and the second best P score. On the LaSOText set, the proposed PJVLT achieves comparable performance with VLTTransT, whereas VLTTransT adopts more training datasets and more complex training steps to train the model to recognize the unseen target. It is worth pointing out that in comparison with the tracking performance on the other datasets, the tracking performance on the LaSOText is much lower. On one hand, it is nontrivial to align the unseen semantics in language descriptions with the unseen targets in visual patches without pre-training. On the other hand, the ambiguous semantics in language descriptions are harmful to discriminate target objects and similar distractors. Nevertheless, the proposed PJVLT with progressive joint vision-language transformer achieves better performance than most competitors on the LaSOText dataset.

GOT10K [71]. The GOT10K test set consists of 180 videos, and it adopts average overlap (AO), success rate at an overlap threshold of 0.50 ($SR_{0.50}$) and success rate at an overlap threshold of 0.75 ($SR_{0.75}$) as the evaluation metrics. Tab. I illustrates the comparison results of our PJVLT, PJVLT-GOT10K (which is trained on the GOT10K training set) and the other state-of-the-art methods. As shown in Tab. I, the proposed PJVLT reaches the state-of-the-art performance in comparison with the other competing methods. To be more specific, our PJVLT achieves the best performance with an AO score of 76.2%, a $SR_{0.50}$ score of 85.8%, and a $SR_{0.75}$ score of 74.2% on the test set. In comparison with the vision-only tracking methods (e.g., STARK, MixFormer, OSTrack, SimTrack, SwinTrack and CTTrack) which are also trained on the GOT10K training set, our model sets a new state-of-the-art with an AO score of 72.8%, a $SR_{0.50}$ score of 83.0%, and a $SR_{0.75}$ score of 70.4% on the test set. This shows that performing progressive joint vision-language representation learning in our PJVLT is an effective way to achieve the state-of-the-art performance. Moreover, our PJVLT is superior to the other vision-language tracking methods (i.e., SNLT, VLTSiamCAR and VLTTransT) with large performance gains. The excellent performance benefits from the semantic-visual alignment and refinement layers in our PJVLT to facilitate progressive joint vision-language encoding.

TNL2K [75]. TNL2K is a recent dataset specially designed for vision-language tracking, where the test set contains 700 videos. The TNL2K test set adopts the same evaluation metrics as LaSOT and LaSOText. Tab. I compares the proposed PJVLT with seven vision-only tracking methods and five vision-language tracking methods. As reported in Tab. I, our PJVLT sets a new state-of-the-art with 56.2% AUC score, 72.9% NP score, and 57.2% P score. Compared to the seven vision-only tracking methods, our PJVLT outperforms the second best vision-only tracking method SimTrack by 1.4/3.4 absolute points on the AUC/P metric. Furthermore, our PJVLT achieves the second best results among the six vision-language tracking methods. The superior performance of JointNLT over PJVLT can be attributed to two aspects. On one hand, JointNLT introduces the additional semantic-guided temporal model to cope with the target appearance variations over time. On the other hand, JointNLT employs additional TNL2K train set to train its model. In contrast, our PJVLT uses the same common datasets as most tracking methods to train a more succinct model. Moreover, our PJVLT runs at a faster tracking speed than JointNLT.

TABLE II
COMPARISONS OF THE THREE SEMANTIC-VISUAL FUSION METHODS ON THE LASOT AND GOT10K TEST SETS. THE PROPOSED PJVLT USING THE PROGRESSIVE FUSION METHOD ACHIEVES THE HIGHEST SCORES, WHICH ARE HIGHLIGHTED BY **BOLD**.

Method	LaSOT			GOT10K		
	AUC	NP	P	AO	$SR_{0.50}$	$SR_{0.75}$
Baseline	64.5	73.8	69.3	71.0	80.4	68.2
Late Fusion	65.2	74.5	70.1	71.8	81.4	68.6
Early Fusion	65.5	74.5	70.3	72.4	82.3	69.8
Progressive Fusion	**65.8**	**75.3**	**71.0**	**72.8**	**83.0**	**70.4**

C. Ablative Experiments

We perform ablative experiments to show the effectiveness of our design in PJVLT. As the LaSOT and GOT10K test sets with diverse classes are appropriate to assess the generalization of the proposed PJVLT, we choose them for our ablative experiments.

Effectiveness of Progressive Joint Vision-Language Encoding. The proposed PJVLT performs progressive joint vision-language representation learning for object tracking. To show the effectiveness of our progressive fusion method, we carefully design our model with the early fuison method and the late fusion method, respectively. Tab. II reports the comparison results of our model using the three fusion methods. As observed from the table, compared to the baseline model, our model with the fusion methods can consistently improve the tracking performance on the test set, which shows that the introduction of language information is effective to facilitate object tracking. Compared to the baseline method, the late fusion method can result in 0.7/0.7/0.8 absolute point gains in terms of the AUC/NP/P score on the LaSOT test set. In comparison with the late fusion method, the early fusion method can yield 0.6/0.9/1.2 absolute point gains on the AO/$SR_{0.50}$/$SR_{0.75}$ metric on the GOT10K test set. However, both late fusion method and early fusion method are inferior to our progressive fusion method, which facilitate our PJVLT model to achieve the best performance with an AO/$SR_{0.50}$/$SR_{0.75}$

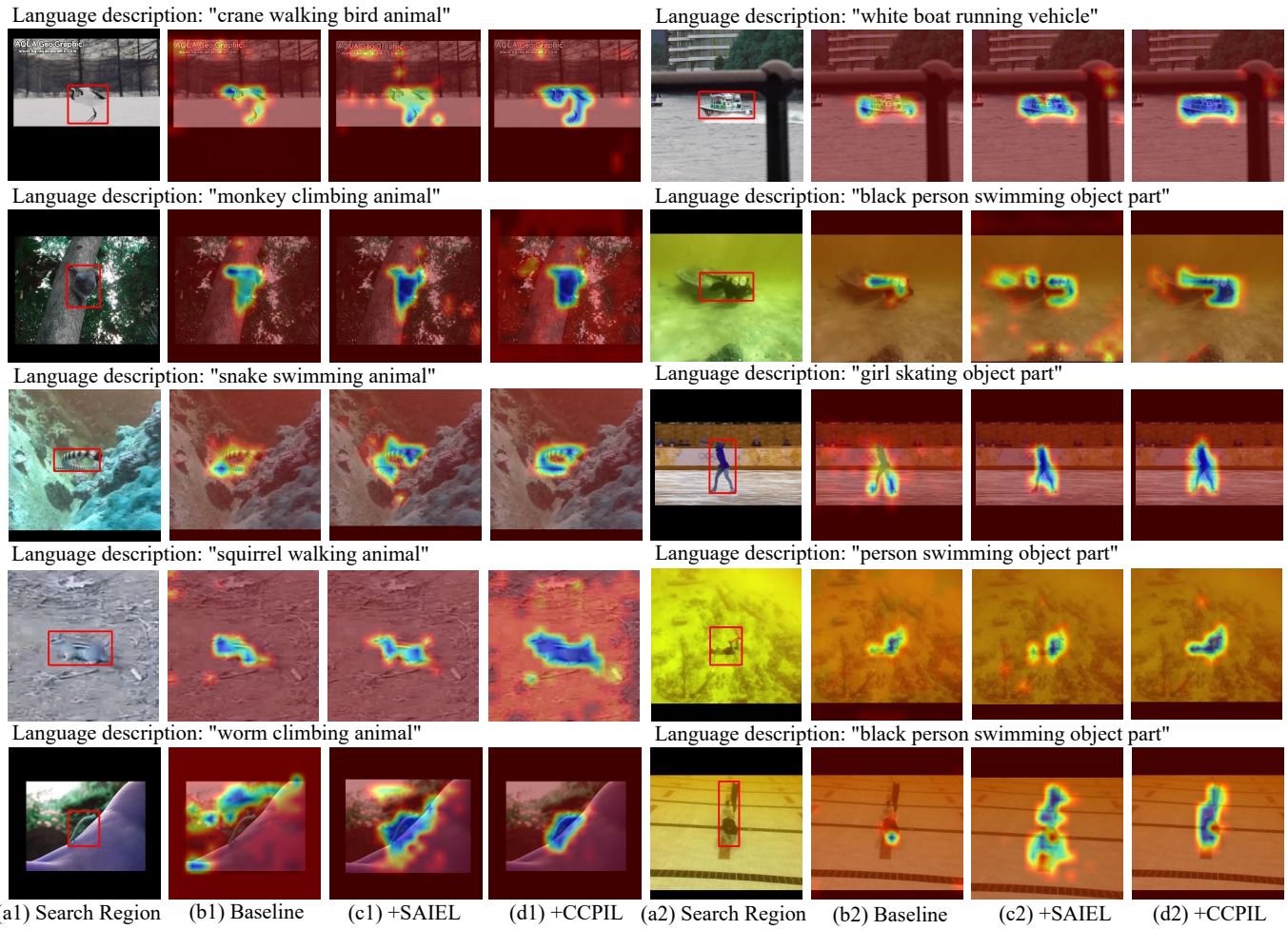


Fig. 6. Comparisons of activation maps of our PJVLT with SAIEL and CCPIL using GradCAM [65] on the GOT10K test set. Both SAIEL and CCPIL can endow the discriminability power of our PJVLT model to distinguish the targets from surronding backgrounds.

TABLE III
IMPACT OF THE NUMBER OF PROGRESSIVE SEMANTIC-VISUAL FUSION LAYERS ON THE LASOT AND GOT10K TEST SETS. THE PROPOSED PJVLT WITH 12 FUSION LAYERS ACHIEVES THE HIGHEST SCORES, WHICH ARE HIGHLIGHTED BY **BOLD**.

Layer	LaSOT			GOT10K		
	AUC	NP	P	AO	$SR_{0.50}$	$SR_{0.75}$
1	64.6	73.9	69.6	72.1	81.9	69.4
3	64.7	74.3	69.7	72.1	81.9	69.6
6	64.7	74.4	69.9	72.3	82.0	69.7
9	65.5	74.5	70.3	72.7	82.6	69.9
12	**65.8**	**75.3**	**71.0**	**72.8**	**83.0**	**70.4**

TABLE IV
INFLUENCE OF SAIEL AND CCPIL ON THE LASOT AND GOT10K TEST SETS. THE PROPOSED PJVLT WITH BOTH SAIEL AND CCPIL ACHIEVES THE HIGHEST SCORES, WHICH ARE HIGHLIGHTED BY **BOLD**.

Method	SAIEL	CCPIL	LaSOT			GOT10K		
			AUC	NP	P	AO	$SR_{0.50}$	$SR_{0.75}$
Baseline			64.5	73.8	69.3	71.0	80.4	68.2
w/o CCPIL	✓		65.4	74.9	70.4	72.0	81.9	68.8
w/o SAIEL		✓	64.5	73.9	69.6	72.0	81.9	69.6
PJVLT	✓	✓	**65.8**	**75.3**	**71.0**	**72.8**	**83.0**	**70.4**

score of 72.8%/83.0%/70.4% on the GOT10K test set and an AUC/NP/P score of 65.8%/75.3%/71.0% on the LaSOT test set.

Impact of the Number of Progressive Semantic-Visual Fusion Layers. The proposed PJVLT adopts 12 joint vision-language encoder layers to performs semantic-visual alignment and refinement. To investigate the influence of the number of progressive semantic-visual fusion layers on the tracking performance, we cautiously design four variants of the proposed PJVLT, which respectively adopts 1, 3, 6 and 9 layers for semantic-visual fusion. Table III reports the tracking performance of these variants on the LaSOT and GOT10K test sets. From Table III, we can observe that as the number of progressive fusion layers increases, the tracking performance of these variants can be gradually improved.

Effectiveness of SAIEL and CCPIL. The core components of the proposed PIVLT are SAIEL and CCPIL. To test the influence of SAIEL and CCPIL on the performance of our PJVLT, we remove them from our PJVLT and report the comparison results in Tab. IV. From Tab. IV, we can observe that both SAIEL and CCPIL are conductive for performance improvement on the test set, which validates the effectiveness of SAIEL and CCPIL. In comparison with the baseline method, SAIEL can yield 0.9/1.1/1.2 absolute point gains for the AUC/NP/P score on the LaSOT test set. In contrast,

TABLE V
COMPARISONS OF DIFFERENT FEATURE INTEGRATION SCHEMES IN SAIEL ON THE LASOT AND GOT10K TEST SETS. OUR PJVLT USING A CONCATENATION OPERATION OBTAINS THE HIGHEST SCORES, WHICH ARE HIGHLIGHTED BY **BOLD**.

Operator	$\oplus$	$\odot$	©	LaSOT			GOT10K		
				AUC	NP	P	AO	$SR_{0.50}$	$SR_{0.75}$
Summation	✓			65.6	74.8	70.6	72.7	82.4	70.0
Multiplication		✓		65.1	74.5	70.3	71.8	81.5	69.0
Concatenation			✓	**65.8**	**75.3**	**71.0**	**72.8**	**83.0**	**70.4**

TABLE VI
INFLUENCE OF CCM AND PIM IN CCPIL ON THE LASOT AND GOT10K TEST SETS. THE PROPOSED PJVLT WITH BOTH CCM AND PIM ACHIEVES THE HIGHEST SCORES, WHICH ARE HIGHLIGHTED BY **BOLD**.

Method	CCM	PIM	LaSOT			GOT10K		
			AUC	NP	P	AO	$SR_{0.50}$	$SR_{0.75}$
w/o CCPIL			65.4	74.9	70.4	72.0	81.9	68.8
w/o CCM		✓	65.7	75.1	70.6	72.2	82.1	69.4
w/o PIM	✓		65.6	75.0	70.9	72.4	82.2	69.7
PJVLT	✓	✓	**65.8**	**75.3**	**71.0**	**72.8**	**83.0**	**70.4**

CCPIL can result in 1.0/1.5/1.4 absolute point gains on the AO/$SR_{0.50}$/$SR_{0.75}$ metric. Overall, compared with the baseline, the proposed PJVLT with both SAIEL and CCPIL can bring 2.5%/3.2%/3.2% and 2.0%/2.0%/2.5% relative performance gains for the AO/$SR_{0.50}$/$SR_{0.75}$ and AUC/NP/P score on the GOT10K and LaSOT test set, respectively.

Comparisons of Different Feature Integration Schemes in SAIEL. The proposed SAIEL employs the concatenation operation to integrate semantic features and visual features as shown in Fig. 4. In Tab. V, we illustrate the comparison results by using different operations, including element-wise summation (i.e., $\oplus$), element-wise multiplication (i.e., $\odot$) and concatenation (i.e., ©). As shown in Tab. V, in comparison with the tracking methods using the element-wise summation or element-wise multiplication operator, our PJVLT using the concatenation operation exhibits the best performance on the GOT10K and LaSOT test sets.

Effectiveness of CCM and PIM Modules in CCPIL. Each CCPIL employs a channel communication module (CCM) and a patch interaction module (PIM) to refine channels and patches of visual-semantic tokens for target-aware multi-modal tracking as depicted in Fig. 5. To test the influence of CCM and PIM on the tracking performance, we remove each component from CCPIL and report the comparison results in Tab. VI. As observed from the table, both CCM and PIM are effective to improve the tracking performance, and the proposed PJVLT using both CCM and PIM achieves the best performance in comparison with its counterparts on the GOT10K and LaSOT test sets.

D. Qualitative Comparison

In this section, we compare the visualization maps of our PJVLT model with/without SAIEL and CCPIL on the GOT10K test set. Fig. 6, Fig. 7 and Fig. 8 show the activation maps of our PJVLT model by using GradCAM [65]. The highest prediction score in the classification map is selected as the target and it is back-propagated to the output of our progressive joint vision-language encoder for visualization. From Fig. 6, Fig. 7 and Fig. 8, we can observe that our PJVLT with both SAIEL and CCPIL can focus on the targets and suppress the backgrounds, validating the effectiveness of progressive joint vision-language representation learning.

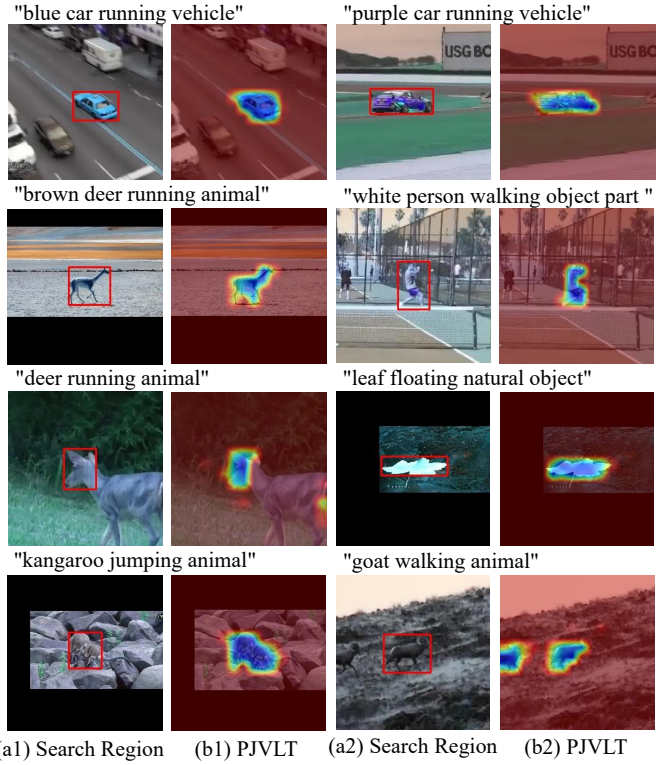
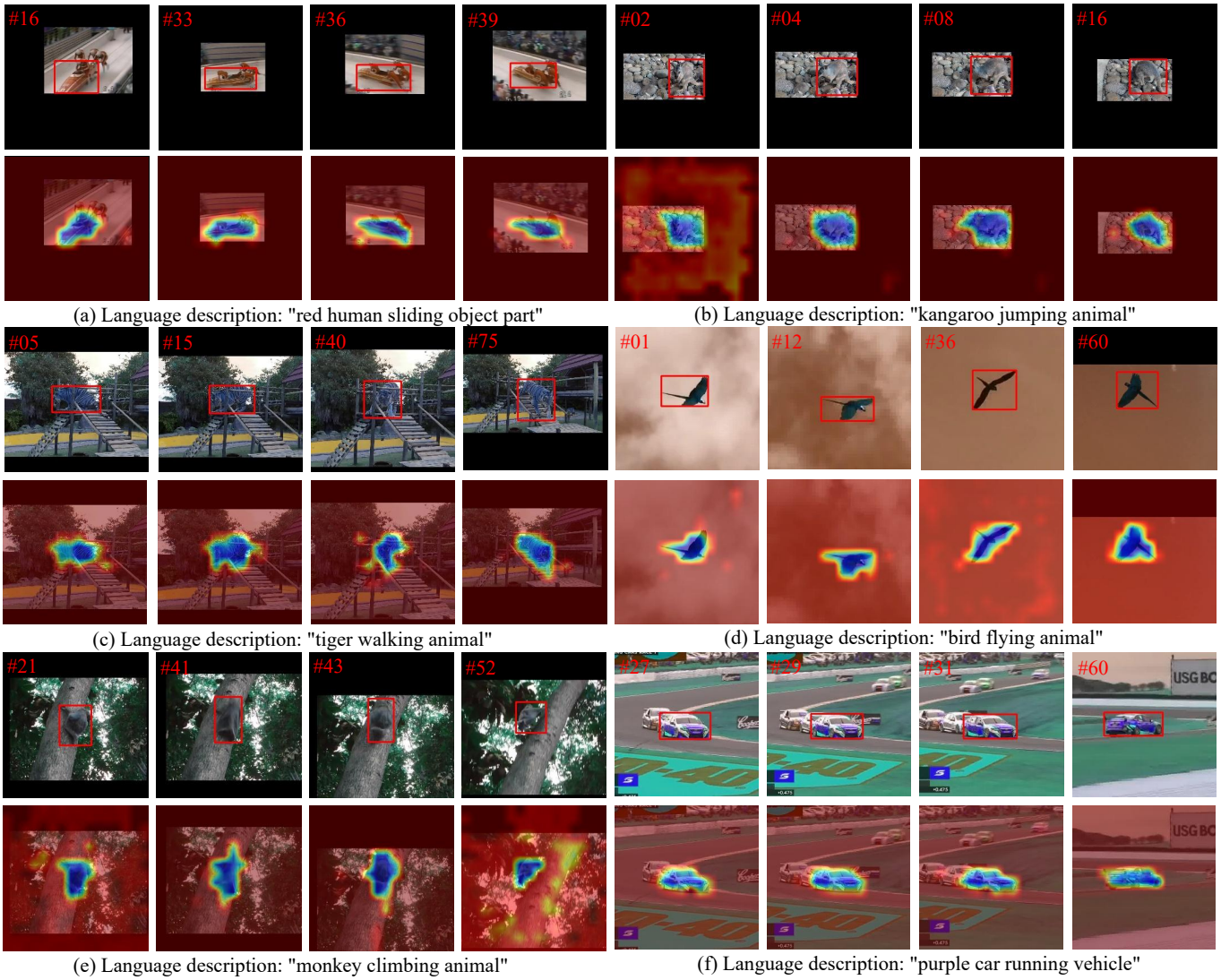


Fig. 7. Activation maps of our PJVLT model on eight examples using GradCAM [65] on the GOT10K test set.

As depicted on the top four rows of the left four columns in Fig. 6, the baseline method without using language descriptions only focuses on relatively small regions of various animals (i.e., "crane", "monkey", "snake" and "squirrel"); when introducing SAIEL, our PJVLT model using attribute descriptions is able to attend to discriminative parts of these animals; when further introducing CCPIL, our PJVLT model using attribute descriptions can precisely concentrate on the whole target regions of these animals. The bottom row of the left four columns in Fig. 6 shows that without language attributes, the baseline method cannot activate the target region of "worm"; by introducing the proposed SAIEL into the baseline model, the responses of the intermediate model scatter on both the target regions and background regions; by further introducing CCPIL into the intermediate model, the responses of our PJVLT model centralize on the whole target region of "worm". On the top four rows of the right four columns in Fig. 6, we can observe that by gradually introducing the proposed SAIEL and CCPIL, our PJVLT model can activate more and more discriminative target regions of "boat" or "person" on the search region. As shown on the bottom row of the right four columns in Fig. 6, the baseline model without language attributes can merely activate the head of the



(a) Language description: "red human sliding object part"

(b) Language description: "kangaroo jumping animal"

(c) Language description: "tiger walking animal"

(d) Language description: "bird flying animal"

(e) Language description: "monkey climbing animal"

(f) Language description: "purple car running vehicle"

Fig. 8. Activation maps of our PJVLT model across various frames of six video sequences using GradCAM [65] on the GOT10K test set.

"black swimming person"; with the introduction of SAIEL, the intermediate model with language attributes coarsely attend to both target regions and background regions. In contrast, with both SAIEL and CCPIL, our PJVLT with language attributes can accurately focus on the discriminative target regions of "black swimming person". In general, the above observation reveals that both SAIEL and CCPIL are beneficial to our PJVLT model for better target localization.

Fig. 7 illustrates eight examples activated by our PJVLT model. As depicted on the top three rows in Fig. 7, when using unambiguous language expressions of various semantics as inputs, the proposed PJVLT model can successfully and precisely activate the discriminative target regions of various objects. However, from the bottom row in Fig. 7, we can observe that when the input language expressions have ambiguous semantics in the search regions, the proposed PJVLT model will activate both target regions and distractor regions. For instance, when two similar "jumping kangaroos" are heavily occluded with each other, the responses of them are fused together. Furthermore, when a "walking goat" is distracted by another "walking goat", the proposed PJVLT model will activate two separate regions of "walking goats". Fig. 8 shows the activation maps that span multiple frames within six video sequences. As shown in Fig. 8(a) and Fig. 8(f), when the target objects undergo significant viewpoint changes, our PJVLT model can precisely activate the target regions from different viewpoints across multiple frames. From Fig. 8(b) to Fig. 8(e), we can observe that our PJVLT model is able to outline various shapes of different animals across multiple frames. The vivid visualization results further prove that our PJVLT model is effective to enhance vision-language context modeling.

V. Conclusions

In this paper, we propose a novel progressive joint vision-language transformer (PJVLT) to perform multi-modal representation learning for vision-language tracking. The proposed PJVLT carefully plugs a semantic-aware instance encoder layer (SAIEL) and a channel communication patch interaction layer (CCPIL) into each intermediate layer of joint vision-language transformer encoder. SAIEL is eligible for alignment between visual features and semantic features from coarse to

fine level, and CCPIL is responsible for activation of target-aware feature patches and channels of aligned multi-modal features. By aligning and refining semantic-visual features at each intermediate layer of transformer encoder, our PJVLT can adaptively excavate well-aligned vision-language context to enhance the target at hierarchical levels. Experiments on four prevalent single object tracking datasets show that our PJVLT can achieve the state-of-the-art performance in comparison with both conventional tracking methods and vision-language tracking methods.

REFERENCES

[1] D. Sun, L. Cheng, S. Chen, C. Li, Y. Xiao, and B. Luo, "Uav-ground visual tracking: A unified dataset and collaborative learning approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 5, pp. 3619–3632, 2024.

[2] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, "Deep learning for visual tracking: A comprehensive survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 3943–3968, 2022.

[3] Y. Liu, Y. Liang, Q. Wu, L. Zhang, and H. Wang, "A new framework for multiple deep correlation filters based object tracking," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 1670–1674.

[4] J. Liao, C. Qi, and J. Cao, "Temporal constraint background-aware correlation filter with saliency map," *IEEE Transactions on Multimedia*, vol. 23, pp. 3346–3361, 2021.

[5] Y. Zheng, Y. Zhang, and B. Xiao, "Target-aware transformer tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 9, pp. 4542–4551, 2023.

[6] Y. Yang and X. Gu, "Joint correlation and attention based feature fusion network for accurate visual tracking," *IEEE Transactions on Image Processing*, vol. 32, pp. 1705–1715, 2023.

[7] G. Chen, P. Zhu, B. Cao, X. Wang, and Q. Hu, "Cross-drone transformer network for robust single object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 9, pp. 4552–4563, 2023.

[8] S. Yao, X. Han, H. Zhang, X. Wang, and X. Cao, "Learning deep lucas-kanade siamese network for visual tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 4814–4827, 2021.

[9] Z. Zhou, X. Zhou, Z. Chen, P. Guo, Q.-Y. Liu, and W. Zhang, "Memory network with pixel-level spatio-temporal learning for visual object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 11, pp. 6897–6911, 2023.

[10] C. Zhuang, Y. Liang, Y. Yan, Y. Lu, and H. Wang, "Bounding box distribution learning and center point calibration for robust visual tracking," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 4718–4722.

[11] L. Jiao, D. Wang, Y. Bai, P. Chen, and F. Liu, "Deep learning in visual tracking: A review," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–20, 2021.

[12] C. Tang, X. Wang, Y. Bai, Z. Wu, J. Zhang, and Y. Huang, "Learning spatial-frequency transformer for visual object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 9, pp. 5102–5116, 2023.

[13] L. Xiong, Y. Liang, Y. Yan, and H. Wang, "Correlation filter tracking with adaptive proposal selection for accurate scale estimation," in *Proceedings of the IEEE IEEE International Conference on Multimedia and Expo (ICME)*, 2019, pp. 1816–1821.

[14] W.-M. Hu, Q. Wang, J. Gao, B. Li, and S. Maybank, "Dcfnet: Discriminant correlation filters network for visual tracking," *Journal of Computer Science and Technology*, 2023.

[15] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, 2016, pp. 850–865.

[16] B. Ye, H. Chang, B. Ma, S. Shan, and X. Chen, "Joint feature learning and relation modeling for tracking: A one-stream framework," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 341–357.

[17] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 6182–6191.

[18] Z. Fu, Q. Liu, Z. Fu, and Y. Wang, "Stmtrack: Template-free visual tracking with space-time memory networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 769–13 778.

[19] Y. Liang, H. Chen, Q. Wu, C. Xia, and J. Li, "Joint spatio-temporal similarity and discrimination learning for visual tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 8, pp. 7284–7300, 2024.

[20] Z. Li, R. Tao, E. Gavves, C. G. M. Snoek, and A. W. M. Smeulders, "Tracking by natural language specification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7350–7358.

[21] Q. Feng, V. Ablavsky, Q. Bai, G. Li, and S. Sclaroff, "Real-time visual object tracking with natural language description," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 689–698.

[22] Q. Feng, V. Ablavsky, Q. Bai, and S. Sclaroff, "Siamese natural language tracker: Tracking by natural language descriptions with siamese trackers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 5847–5856.

[23] M. Guo, Z. Zhang, H. Fan, and L. Jing, "Divert more attention to vision-language tracking," in *Proceedings of the 35th Conference on Neural Information Processing Systems (NeuraIPS)*, 2022, pp. 4446–4460.

[24] L. Zhou, Z. Zhou, K. Mao, and Z. He, "Joint visual grounding and tracking with natural language specification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 23 151–23 160.

[25] Y. Zheng, B. Zhong, Q. Liang, G. Li, R. Ji, and X. Li, "Toward unified token learning for vision-language tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 4, pp. 2125–2135, 2024.

[26] Y. Shao, S. He, Q. Ye, Y. Feng, W. Luo, and J. Chen, "Context-aware integration of language and visual references for natural language tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 19 208–19 217.

[27] R. Wang, Z. Tang, Q. Zhou, X. Liu, T. Hui, Q. Tan, and S. Liu, "Unified transformer with isomorphic branches for natural language tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 9, pp. 4529–4541, 2023.

[28] Y. Liang, Q. Wu, Y. Liu, Y. Yan, and H. Wang, "Correlation filter tracking with shepherded instance-aware proposals," in *Proceedings of the 28th ACM International Conference on Multimedia (MM)*, 2018, pp. 420–428.

[29] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8971–8980.

[30] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 7183–7192.

[31] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4293–4302.

[32] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1349–1358.

[33] I. Jung, J. Son, M. Baek, and B. Han, "Real-time mdnet," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 89–104.

[34] Y. Liang, Y. Liu, Y. Yan, L. Zhang, and H. Wang, "Robust visual tracking via spatio-temporal adaptive and channel selective correlation filters," *Pattern Recognition*, vol. 112, pp. 107 738:1–107 738:14, 2021.

[35] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6931–6939.

[36] Y. Liang, Q. Wu, Y. Liu, Y. Yan, and H. Wang, "Deep correlation filter tracking with shepherded instance-aware proposals," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 408–11 421, 2022.

[37] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines." in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 12 549–12 556.

[38] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4282–4291.

[39] Y. Liang, P. Zhao, Y. Hao, and H. Wang, "Siamese template diffusion networks for robust visual tracking," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2022, pp. 1–6.

[40] N. Wang, W. Zhou, Q. Tian, and H. Li, "Cascaded regression tracking: Towards online hard distractor discrimination," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1580–1592, 2021.

[41] N. Wang, W. Zhou, and H. Li, "Learning diverse models for end-to-end ensemble tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 2220–2231, 2021.

[42] B. Yu, M. Tang, L. Zheng, G. Zhu, J. Wang, H. Feng, X. Feng, and H. Lu, "High-performance discriminative tracking with transformers," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 9836–9845.

[43] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1571–1580.

[44] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8122–8131.

[45] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 10 428–10 437.

[46] C. Mayer, M. Danelljan, G. Bhat, M. Paul, D. P. Paudel, F. Yu, and L. Van Gool, "Transforming model prediction for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8731–8740.

[47] Z. Song, J. Yu, Y.-P. P. Chen, and W. Yang, "Transformer tracking with cyclic shifting window attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8791–8800.

[48] Y. Cui, C. Jiang, L. Wang, and G. Wu, "Mixformer: End-to-end tracking with iterative mixed attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 13 608–13 618.

[49] S. Gao, C. Zhou, and J. Zhang, "Generalized relation modeling for transformer tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 18 686–18 695.

[50] Z. Song, R. Luo, J. Yu, Y.-P. P. Chen, and W. Yang, "Compact transformer tracker with correlative masked modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 37, no. 2, 2023, pp. 2321–2329.

[51] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 213–229.

[52] Q. Wu, T. Yang, Z. Liu, B. Wu, Y. Shan, and A. B. Chan, "Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 14 561–14 571.

[53] X. Wei, Y. Bai, Y. Zheng, D. Shi, and Y. Gong, "Autoregressive visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 9697–9706.

[54] Y. Liao, A. Zhang, Z. Chen, T. Hui, and S. Liu, "Progressive language-customized visual feature learning for one-stage visual grounding," *IEEE Transactions on Image Processing*, vol. 31, pp. 4266–4277, 2022.

[55] R. Hong, D. Liu, X. Mo, X. He, and H. Zhang, "Learning to compose and reason with language tree structures for visual grounding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 684–696, 2022.

[56] Z. Yang, J. Wang, Y. Tang, K. Chen, H. Zhao, and P. H. Torr, "Lavt: Language-aware vision transformer for referring image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 18 134–18 144.

[57] S. Liu, T. Hui, S. Huang, Y. Wei, B. Li, and G. Li, "Cross-modal progressive comprehension for referring segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4761–4775, 2022.

[58] J. Wu, Y. Jiang, P. Sun, Z. Yuan, and P. Luo, "Language as queries for referring video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4964–4974.

[59] A. Botach, E. Zheltonozhskii, and C. Baskin, "End-to-end referring video object segmentation with multimodal transformers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4975–4985.

[60] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[61] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019, pp. 4171–4186.

[62] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[63] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023, pp. 4195–4205.

[64] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Re, "Flashattention: Fast and memory-efficient exact attention with IO-awareness," in *Proceedings of the Advances in Neural Information Processing Systems (NeuraIPS)*, 2022, pp. 16 344–16 359.

[65] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[66] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[67] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 765–781.

[68] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666.

[69] K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 979–15 988.

[70] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5374–5383.

[71] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1562–1577, 2021.

[72] M. Müller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 310–327.

[73] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 740–755.

[74] H. Fan, H. Bai, L. Lin, F. Yang, and H. Ling, "Lasot: A high-quality large-scale single object tracking benchmark," *International Journal of Computer Vision*, vol. 129, no. 8, 2021.

[75] X. Wang, X. Shu, Z. Zhang, B. Jiang, Y. Wang, Y. Tian, and F. Wu, "Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 758–13 768.

[76] P. Voigtlaender, J. Luiten, P. H. Torr, and B. Leibe, "Siam r-cnn: Visual tracking by re-detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6577–6587.

[77] K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, and X. Yang, "High-performance long-term tracking with meta-updater," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6297–6306.
[78] Z. Zhang, Y. Liu, X. Wang, B. Li, and W. Hu, "Learn to match: Automatic matching network design for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 13 319–13 328.
[79] Y. Cui, D. Guo, Y. Shao, Z. Wang, C. Shen, L. Zhang, and S. Chen, "Joint classification and regression for visual tracking with fully convolutional siamese networks," *International Journal of Computer Vision*, vol. 130, pp. 550–566, 2022.
[80] F. Ma, M. Z. Shou, L. Zhu, H. Fan, Y. Xu, Y. Yang, and Z. Yan, "Unified transformer tracker for object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8771–8780.
[81] B. Chen, P. Li, L. Bai, L. Qiao, Q. Shen, B. Li, W. Gan, W. Wu, and W. Ouyang, "Backbone is all your need: A simplified architecture for visual object tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 375–392.
[82] L. Lin, H. Fan, Z. Zhang, Y. Xu, and H. Ling, "Swintrack: A simple and strong baseline for transformer tracking," in *Proceedings of the 35th Conference on Neural Information Processing Systems (NeuraIPS)*, 2022, pp. 16 743–16 754.

Yanjie Liang received the Ph.D. degree in computer science with the School of Informatics, Xiamen University, Xiamen, China, in 2021. He is currently an assistant professor at Pengcheng Laboratory. He has published several papers in IEEE TIP, IEEE TITS, IEEE TCSVT, Pattern Recognition, ACM MM, ICME and ICASSP. His current research interests include computer vision, machine learning, and visual tracking.

Qiangqiang Wu received the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has published several articles in IEEE TPAMI, IEEE TIE, CVPR, ICCV, AAAI, ACM MM, ICME and ICASSP. His current research interests include computer vision, deep learning, adversarial learning, and visual tracking.

Lin Cheng received the Ph.D. degree in computer science with the School of Informatics, Xiamen University, Xiamen, China, in 2023. His current research interests include computer vision, machine learning, and deep learning theory. He serves as a reviewer of CVPR and ICCV.

Changqun Xia received the Ph.D. degree from the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China, in 2019. He is currently an Associate Professor with Peng Cheng Laboratory. He has published several articles in IEEE TPAMI, IEEE TIP, CVPR, ICCV, AAAI and ACM MM. His research interests include computer vision and image understanding.

Jia Li is currently a Full Professor with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University. He received his B.E. degree from Tsinghua University in 2005 and Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences, in 2011. Before he joined Beihang University in 2014, he used to work at Nanyang Technological University, Shanda Innovations, and Peking University. His research interests mainly cover the visual perception and understanding of extreme environments. His research work has been supported by the NSFC Key Project and the NSFC Excellent Young Researcher Fund. He is the author or co-author of more than 120 technical articles in refereed journals and conferences such as IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal of Computer Vision, CVPR, and ICCV. He also has more than 70 patents issued from U.S. and China. He is now an Associate Editor of IET CV and IEEE MM. He was also selected into the Beijing Nova Program and ever received the Second-grade Science Award from the Chinese Institute of Electronics (2018), two Excellent Doctoral Thesis Awards from the Chinese Academy of Sciences (2012) and the Beijing Municipal Education Commission, and the First-Grade Science Technology Progress Award from Ministry of Education, China. He is a Fellow of IET, a Distinguished Member of CCF, and a senior member of ACM and CIE.