ATCTrack: Aligning Target-Context Cues with Dynamic Target States for Robust Vision-Language Tracking

Xiaokun Feng[1,2*] **Shiyu Hu**[3*] **Xuchen Li**[1,2,4] **Dailing Zhang**[1,2]
Meiqi Wu[2] **Jing Zhang**[2] **Xiaotang Chen**[1,2] **Kaiqi Huang**[1,2 †]
[1]School of Artificial Intelligence, UCAS
[2]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems, CASIA
[3]School of Physical and Mathematical Sciences, NTU [4]ZGCA

Abstract

*Vision-language tracking aims to locate the target object in the video sequence using a template patch and a language description provided in the initial frame. To achieve robust tracking, especially in complex long-term scenarios that reflect real-world conditions as recently highlighted by MGIT, it is essential not only to characterize the target features but also to utilize the context features related to the target. However, the visual and textual target-context cues derived from the initial prompts generally align only with the initial target state. Due to their dynamic nature, target states are constantly changing, particularly in complex long-term sequences. It is intractable for these cues to continuously guide Vision-Language Trackers (VLTs). Furthermore, for the text prompts with diverse expressions, our experiments reveal that existing VLTs struggle to discern which words pertain to the target or the context, complicating the utilization of textual cues. In this work, we present a novel tracker named **ATCTrack**, which can obtain multimodal cues **A**ligned with the dynamic target states through comprehensive **T**arget-**C**ontext feature modeling, thereby achieving robust tracking. Specifically, **(1)** for the visual modality, we propose an effective temporal visual target-context modeling approach that provides the tracker with timely visual cues. **(2)** For the textual modality, we achieve precise target words identification solely based on textual content, and design an innovative context words calibration method to adaptively utilize auxiliary context words. **(3)** We conduct extensive experiments on mainstream benchmarks and ATCTrack achieves a new SOTA performance. The code and models will be released at: https://github.com/XiaokunFeng/ATCTrack.*

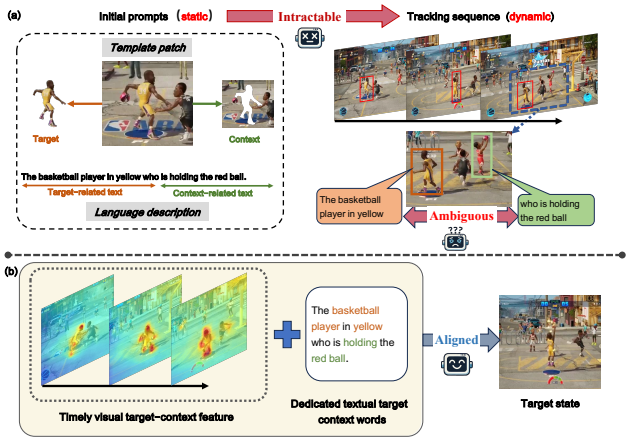

Figure 1. **(a) Limitations of initially given prompts** (*i.e.*, the template patch and language description). Despite containing certain multimodal target-context cues, these static initial prompts are intractable for continuously guiding the tracker in dynamic tracking sequences. Particularly, the objects corresponding to the target-related text and context-related text can be ambiguous, which may mislead the tracker. **(b) Our key insights lie in providing multimodal target-context cues aligned with the dynamic target states.** For the visual modality, we model timely visual target-context features to adapt to dynamic changes. For the textual modality, we achieve precise awareness of target words and mitigate the potential misleading effects of context words.

1. Introduction

Given a template patch and a language description in the initial frame, Vision-Language Tracking (**VLT**) task [49] aims to locate a user-defined object in a video sequence. Harnessing the complementary advantages of multiple modalities [40], recent studies, exemplified by MGIT [33], seek to explore the performance of Vision-Language Trackers (**VLTs**) in **complex long-term** sequences. These scenarios, encompassing multifaceted spatiotemporal and causal

*Equal Contribution
†Corresponding authors

relationships, more accurately reflect real-world conditions while posing new challenges for tracker design.

To achieve robust tracking within these environments, we first need to utilize the provided visual and textual cues to characterize the **target feature** [48]. As shown in Fig. 1 (a), the appearance of the target in the template image and the target-related text (*e.g.*, target class, appearance attributes) in the language description offer basic reference information for tracking. However, complex long-term scenarios are more commonly accompanied by challenges such as occlusion and similar object distractions [35, 92], making it insufficient to rely solely on these target cues. It is also essential to model the **context feature** of the target [65, 81]. Specifically, the visual content surrounding the target and context-related text (*e.g.*, other reference objects) contribute to more robust tracking. Therefore, the target and context features collectively depict the **target states**, and ***efficiently representing and leveraging target-context features*** is crucial for handling complex long-term scenarios.

Considering that the initially given prompts carry certain multimodal target-context information, most existing VLTs, *e.g.*, VLT_{TT} [30] and JointNLT [98], rely entirely on them for tracking. For the visual template image cropped from the first frame, its cropping region is determined by enlarging the target's bounding box, thus including some context features [93]. For the textual description, the full sentence is utilized as a whole. Although achieving some effectiveness, they overlook the inherent dynamic nature of the target states. In complex long-term scenarios, target states often undergo significant changes. As shown in Fig. 1 (a), the subsequent target states gradually deviate from the initially given prompts. This misalignment makes the latter **intractable** for continuously guiding the tracker. Furthermore, for the text prompts, the object corresponding to the target-related text and context-related text can be **ambiguous**, which may significantly mislead the tracker [33, 44].

Recently, some VLTs, *e.g.*, QueryNLT [65] and TTCTrack [58], attempt to handle different word components of the text prompts specifically, giving sufficient attention to target words and alleviating interference from context words. Although well-motivated, these efforts require trackers to automatically distinguish target and context words through visual-textual feature similarity. Due to the lack of supervision information and the diverse nature of textual expressions, we find that this method does not yield satisfactory results. For an intuitive understanding, Fig. 2 (b) and (c) illustrate cases of correct and incorrect awareness of the target words using this method (see Appendix B.1 for implementation details). Additionally, Fig. 2 (a) presents the quantitative evaluation (see Appendix B.2) results, showing that the classification accuracy for target words achieved by this method is only 29.9%.

To address the above issues, we propose a novel tracker named ***ATCTrack***, which comprehensively models **T**arget-**C**ontext features to obtain multimodal cues **A**ligned with the dynamic target states, thereby achieving robust tracking. Our key insights are illustrated in Fig. 1 (b). For the visual modality, we design an effective temporal visual target-context modeling approach to provide the tracker with timely visual cues. Specifically, we explicitly construct a target-context spatial distribution map and integrate it into the updated temporal memory.

For the textual modality, we first propose a precise target words awareness method based solely on textual content. Intuitively, even without relying on video data, we can identify target words purely from the text. For example, we can determine that the target words in the text prompt of Fig. 1 (a) are "basketball player, yellow." Compared to existing methods that rely on fine-grained alignment between textual words and visual targets [58, 65], our approach simplifies the task and achieves an impressive 96.7% target words classification accuracy (Fig. 2). Given the lack of sentence component labels in existing benchmarks [19, 33, 49, 76], we design an automated target words annotation pipeline by leveraging the off-the-shelf Large Language Models (**LLMs**) [2, 70] to provide supervisory labels for model training. Furthermore, leveraging the accurately identified target words, we design an innovative context words calibration mechanism to alleviate the potential misleading impact of context words cues.

Benefiting from these multimodal target-context cues aligned with dynamic target states, ATCTrack significantly outperforms existing SOTAs on mainstream benchmarks (*i.e.*, MGIT [33], TNL2K [76] and LaSOT_{ext} [20]). Impressively, ATCTrack-B improves over the existing best results by 6.4%, 4.3% and 3.5% in precision, respectively. Our contributions are as follows:

- We propose a novel tracker named ATCTrack, which can obtain multimodal cues aligned with the dynamic target states through comprehensive target-context feature modeling. Compared to the limitations of initial prompts, which typically only align with the initial target state, these aligned cues can guide tracking more robustly.
- For the visual modality, ATCTrack effectively models temporal visual target-context features to capture timely visual cues. For the textual modality, ATCTrack achieves precise awareness of target words and mitigates the potential misleading effects of context words.
- We conduct extensive experiments on mainstream benchmarks and ATCTrack achieves a new SOTA performance.

2. Related Works

2.1. Traditional Vision-Language Tracking

Visual language tracking extends the classic visual single object tracking task by incorporating textual descriptions

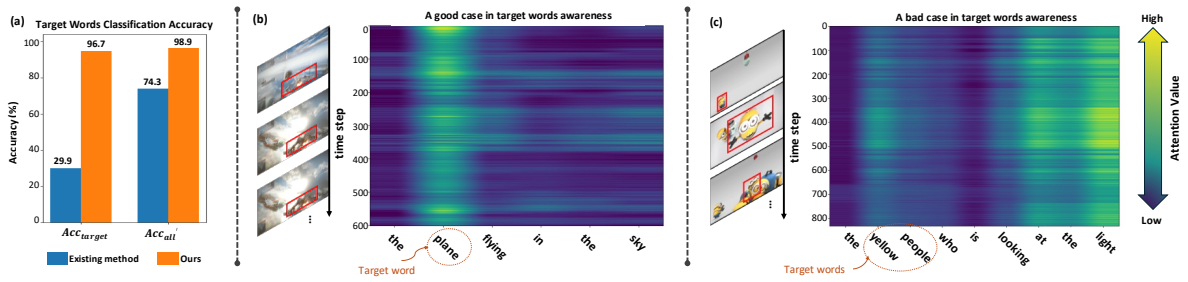

Figure 2. **(a)** Comparison of the existing vision-text similarity-based method and ours in terms of target words classification accuracy. **(b-c)** Attention distribution maps for the target words during tracking using the existing method. In case (b), the model focuses on the word that corresponds to the tracking target, *i.e.*, 'plane,' indicating that the tracker correctly understands the intent embedded in the text prompt, thereby effectively utilizing textual cues. In case (c), the model's focus is on the words 'the light,' which does not match the tracking target 'yellow people,' potentially leading to misguidance in the tracking process. Better viewed in color with zoom-in.

[49]. Many early efforts treat initial prompts as core reference information, utilizing the multimodal target-context cues they contain to guide tracking. Among them, SNLT [23] employs a general language region proposal network to achieve the interaction between multimodal cues and search features, and then uses an aggregation module to integrate multi-scale feature information. To enhance the tracker's ability to align visual and textual modalities, both All-in-One [87] and UVLTrack [56] design specialized contrastive losses [68] to improve the model's multimodal understanding capability. While these VLTs achieve some success, they overly rely on static initial cues and overlook the dynamic nature of the target and its context. This poses a challenge for maintaining robust tracking when the target states deviate from the initial cues [25].

2.2. Object-Context Modeling in Tracking

To accommodate the dynamic changes of target states, many trackers attempt to utilize temporal features [88, 95, 96] to obtain updated target-context cues. For the visual modality, GTI [85] integrates tracking and grounding tasks, replacing the template with updated grounding results. To support longer and denser temporal modeling, some trackers store multiple-step visual temporal features as additional memory. A representative approach involves using RoI features [62] based on predicted bboxes as memory units, *e.g.*, JointNLT [98] and TrDiMP [73]. Given that this localized cropping method can only capture limited visual context information, we represent the global target distribution map and use it to construct temporal visual target-context cues.

In addition to visual modality, recent trackers focus on the unique VLT challenges of misalignment between static textual cues and dynamic target states. MemVLT [25], inspired by prompt learning [1, 42, 97], compresses dynamic target features into a small set of tokens and uses them to implicitly modulate static multimodal cues. To more explicitly leverage multimodal cues, QueryNLT [65] approaches the problem from a target-context perspective, aiming to obtain accurate cues through mutual modulation between dynamic visual features and textual cues. Although QueryNLT shares similar motivations with our work, its key modulation process requires fine-grained alignment between visual targets and textual words in VLTs [89]. As shown in Fig. 2 (a), we conduct a quantitative evaluation and find that this method fails to effectively identify target and context words. In contrast, we design a accurate target words awareness method directly based on textual content and introduce an effective calibration mechanism to mitigate the misleading effects of context words.

3. Methodology

The framework of ATCTrack is depicted in Fig. 3. Given multimodal cues and the search image, the **Text Encoder** and **Vision Encoder** first encode them into specific feature spaces. Then, the **Textual Target-Context Guidance Module** and **Visual Target-Context Guidance Module** sequentially model comprehensive textual and visual target-context features aligned with the dynamic search target, and embed them into the search features. During this process, the **Memory Storage Module** (**MSM**) provides stored visual memory features and saves the updated memory information. Finally, the **Prediction Head** is used to obtain the tracking result based on embedded search features. In the following sections, we will introduce each module in detail.

3.1. Input Encoder

Vision encoder. At time step t, our visual input consists of a search image $x_t \in \mathbb{R}^{3\times H_x\times W_x}$, an initial template patch $z_0 \in \mathbb{R}^{3\times H_z\times W_z}$, and a dynamic template patch $z_t \in \mathbb{R}^{3\times H_z\times W_z}$ updated based on the tracking results [72, 83]. We adopt the one-stream network encoding paradigm [86], which has been widely used by recent mainstream trackers [32, 94, 98]. Specifically, x_t, z_0, and z_t are first projected into token embeddings. Furthermore, inspired by MemVLT [25], we introduce a learnable [CLS] token to capture global visual semantic features [17]. This token is concatenated with template-search tokens and fed into transformer lay-

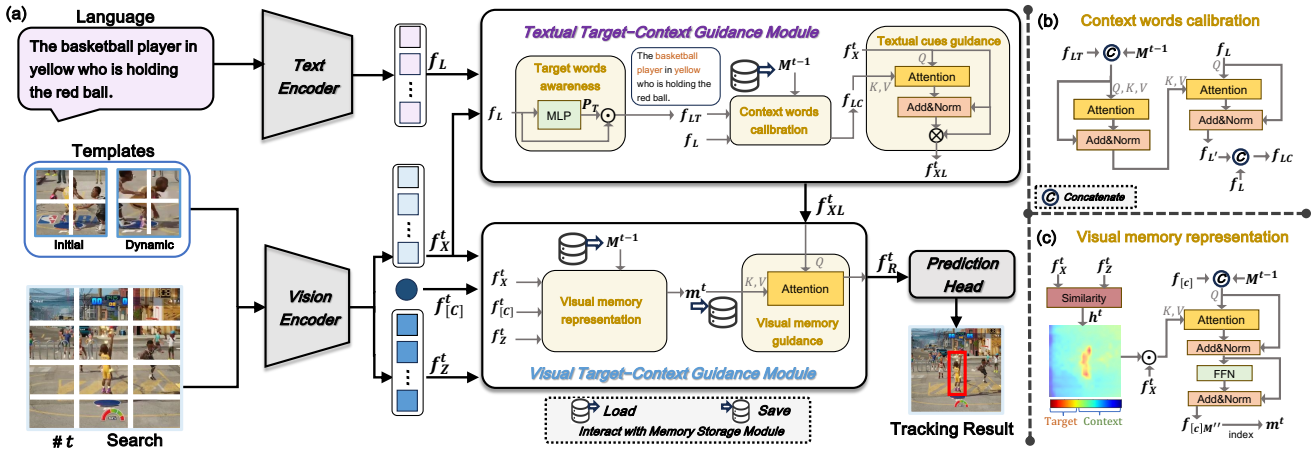

Figure 3. **(a) Framework of our proposed ATCTrack.** Given the language description and template patches as references, ATCTrack locates the target in the search image at time t. The input is first encoded using ***Text*** and ***Vision Encoders***. Subsequently, the ***Textual and Visual Target-Context Guidance Modules*** sequentially embed the aligned textual and visual cues into the search features. During this process, the ***Memory Storage Module*** **(MSM)** provides previously stored memory information (up to $t-1$) and saves updated memory information. Then, the ***Prediction Head*** generates the final tracking results based on the embedded search features. **(b-c) Specific model structures for context words calibration and visual memory representation.**

ers for feature extraction and relational modeling. Finally, we obtain the encoded search feature $f_X^t \in \mathbb{R}^{N_x \times D}$, template feature $f_Z^t \in \mathbb{R}^{N_z \times D}$ corresponding to two template images, and the aggregated [CLS] token $f_{[C]}^t \in \mathbb{R}^{1 \times D}$.

Text encoder. We utilize RoBERTa [17, 52], a classic pre-trained model, as our text encoder. Specifically, for a given sentence, we tokenize it into a sequence of text tokens. The token sequences then are fed into the transformer layers to extract the text embedding feature $f_L \in \mathbb{R}^{N_l \times D}$.

3.2. Textual Target-Context Guidance Module

To fully leverage textual cues, the key insight of this module is to carefully modulate the initial text prompt to adequately focus on the target words and mitigate the potential misleading effects of context words. As shown in Fig. 3, we first explicitly identify the target words from the text content. Then, we design a context words calibration mechanism that leverages the identified target words and the visual memory features provided by the Memory Storage Module (**MSM**) to effectively utilize context words. Finally, we integrate the modulated text features with the search features to achieve the guidance of the textual cues.

Target words awareness. Compared to context words, the information in target words is usually constant and can serve as consistent cues for tracking. However, the diverse nature of textual expressions makes accurately identifying target words challenging. Existing VLTs, such as QueryNLT [65] and TTCTrack [58], attempt to distinguish them based on vision-text similarity. This requirement for fine-grained multimodal alignment increases the learning difficulty. Intuitively, we can identify the target words solely based on the text content. Therefore, we simplify the identification of target words into a multi-label binary classification task, determining whether each word in the sentence belongs to the target words.

Considering the lack of relevant label in existing benchmarks [19, 33, 49, 76], we design specific prompts and leverage the powerful text comprehension capabilities of LLMs [2, 70] to construct an automated and reliable target words annotation pipeline. Due to space constraints, the detailed construction process is described in Appendix A. Leveraging this tailored target words labeling information, we find that accurate target words awareness can be achieved using a lightweight multilayer perceptron built upon the initial text features f_L.

$$p_T = MLP(f_L), \tag{1}$$

where $p_T \in [0,1]^{N_l \times 1}$ denotes the probability of each text token being a target word. By weighting f_L with p_T, we obtain the target words feature $f_{LT} \in \mathbb{R}^{N_l \times D}$.

Context words calibration. The alignment between context words and target states determines whether they guide or mislead the tracker. An intuitive way to determine the timing for utilizing context words is to assess them based on the dynamic evolution of the target states [25, 65]. Therefore, this modeling process requires the temporal memory features stored by the MSM, which represent the latest visual target-context information (details will be described later). Additionally, since context information depends on the target's location, accurately perceiving the target features helps to capture precise context information [4, 9].

Based on these insights, the core of our context words calibration mechanism is to first enhance the target features' representation by integrating identified target words and visual temporal features. These enhanced features are then used to modulate the initial text features to adaptively utilize the context words.

At the current time t, the MSM stores L_m memory units up to time step $t-1$, denoted as $M^{t-1} = \{m_i\}_{i=1}^{L_m}$. As shown in Fig. 3 (b), we first enhance the target feature representation by leveraging the complementarity between the visual memory features M^{t-1} and the textual target words f_{LT}. Specifically, we concatenate the two features:

$$f_{LM} = [f_{LT}; M_*^{t-1}]. \tag{2}$$

Where [;] denotes concatenation along the first dimension, and $M_*^{t-1} \in \mathbb{R}^{L_m \times D}$ is the concatenation of elements in M^{t-1}. Considering the transformer layers' exceptional ability in modeling feature interactions, we apply the vanilla transformer attention operation [71] to f_{LM}:

$$f_{LM'} = Norm(f_{LM} + \Phi_{CA}(f_{LM}, f_{LM})). \tag{3}$$

Here, $\Phi_{CA}(\cdot,\cdot)$ denotes the cross-attention operation, where the first element serves as the query Q and the second element is used to obtain the key K and value V [71]. $Norm$ represents the layer normalization operation.

With the enhancement of target features, the surrounding context information is also indirectly perceived more accurately. Therefore, we use $f_{LM'}$ to modulate the initial static text f_L, allowing for adaptive calibration of the context words. Through the following attention operation, we obtain the meticulously calibrated text features $f_{L'} \in \mathbb{R}^{N_l \times D}$:

$$f_{L'} = Norm(f_L + \Phi_{CA}(f_L, f_{LM'})). \tag{4}$$

Textual cues guidance. Inspired by recent generation models [18, 38, 55] that concatenate various types of text features for utilization, we treat the initial text and the calibrated text as two different types and concatenate them to obtain the comprehensive text feature $f_{LC} = [f_L; f_{L'}]$.

Then, we employ transformer-based cross-attention operations and residual multiplication [94] to obtain the visual search features $f_{XL}^t \in \mathbb{R}^{N_x \times D}$ embedded with textual cues.

3.3. Visual Target-Context Guidance Module

As shown in Fig. 3, this module consists of two core processes: visual memory representation and guidance, aiming to model and leverage dynamic visual target-context memories aligned with the target states. Compared with the sparse dynamic template [82], visual memories can provide denser temporal features. These two mechanisms jointly offer comprehensive temporal information for the tracker. For visual memory representation, we explicitly construct the target-context distribution map and model the memory features at different time steps. Subsequently, we embed the memory cues stored across multiple time steps into the search features, thereby guiding the tracking process.

Visual memory representation. For the [CLS] token $f_{[C]}^t$ encoded by the vision encoder, since it participates in the entire feature integration process of visual information, it serves as a suitable global visual feature representation [17, 25, 88]. Thus, an intuitive approach is to treat $f_{[C]}^t$ as the memory representation at the current time step. However, since the vision encoder is part of the model's early feature modeling stage [86], the precise target location may not have been sufficiently perceived, meaning that $f_{[C]}^t$ lacks explicit awareness of the target-context distribution information. Therefore, we attempt to explicitly construct the target-context distribution heatmap and embed it into $f_{[C]}^t$ to obtain our visual memory.

Specifically, we construct the target-context distribution heatmap by calculating the similarity between the encoded search feature f_X^t and the template features f_Z^t [86]:

$$h^t = (f_X^t \cdot (f_Z^t)^T).mean(dim = 1). \tag{5}$$

Since the f_Z^t are centered on the target, $h^t \in \mathbb{R}^{N_x \times 1}$ reflects the probability of each search token belonging to the target (*i.e.*, not being part of the context). The heatmaps in Fig. 3 (c) are obtained by reshaping h^t into a two-dimensional image space, demonstrating its high interpretability in indicating the spatial distribution of the target-context.

Next, h^t is embedded into $f_{[C]}^t$ to construct the current memory unit m^t. To provide richer target-context dynamic information when constructing m^t, we introduce the historical memory M^{t-1} stored in MSM up to step $t-1$. Note that the L_m memory units in M are obtained through iterative storage of m^t at different time steps. Assuming M^{t-1} has been obtained, we will elaborate on the generation process of each memory unit in the following, while the updating mechanism of M will be presented in Sec. 3.4.

For implementation, we employ the vanilla cross-attention mechanism [71]. In particular, we concatenate the features of $f_{[C]}^t$ and M_*^{t-1} to obtain $f_{[C]M}$ as the query, while using the h^t-weighted f_x^t as both key and value:

$$f_{[C]M'} = Norm(f_{[C]M} + \Phi_{CA}(f_{[C]M}, h^t \odot f_X^t)), \tag{6}$$

$$f_{[C]M''} = Norm(f_{[C]M'} + FFN(f_{[C]M'})). \tag{7}$$

Where $\odot$ represents the Hadamard product, and FFN denotes the feed-forward network. Based on the concatenation index, we extract the feature corresponding to $f_{[C]}^t$ from $f_{[C]M''}$ and utilize it as the memory unit m^t for the current time step. This establishes our memory unit generation mechanism. The generated m^t is then stored in the MSM for subsequent time step computations (see Sec. 3.4).

Method	MGIT (Action)			TNL2K			LaSOT			LaSOT$_{ext}$		
	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P
Basic Variants												
JointNLT [98]	61.0	78.6	44.5	56.9	73.6	58.1	60.4	69.4	63.6	-	-	-
DecoupleTNL [54]	-	-	-	56.7	-	56.0	71.2	-	75.3	-	-	-
All-in-One [87]	-	-	-	55.3	-	57.2	71.7	82.4	78.5	54.5	63.5	-
MMTrack [94]	-	-	-	58.6	75.2	59.4	70.0	82.3	75.7	49.4	59.9	55.3
QueryNLT [65]	-	-	-	56.9	73.6	58.1	59.9	69.6	63.5	-	-	-
OneTracker [32]	-	-	-	58.0	-	59.1	70.5	79.9	76.5	-	-	-
UVLTrack-B [56]	-	-	-	62.7	-	65.4	69.4	-	74.9	49.2	-	55.8
CTVLT [24]	69.2	-	62.9	62.2	-	79.5	72.3	-	79.7	-	-	-
ChatTracker-B [67]	-	-	-	59.6	76.3	62.1	71.7	80.9	77.5	-	-	-
MemVLT [25]	69.4	81.3	63.7	63.3	80.9	67.4	72.9	85.7	80.5	52.1	63.3	59.8
SUTrack-B224 [14]	-	-	-	65.0	-	67.9	73.2	83.4	80.5	53.1	64.2	60.5
SUTrack-B384 [14]	-	-	-	65.6	-	69.3	74.4	83.9	81.9	52.9	63.6	60.1
ATCTrack-B	73.7	84.5	70.1	67.5	85.3	73.6	74.6	87.0	82.1	54.6	65.7	62.8
Performance-oriented Variants												
ChatTracker-L [67]	-	-	-	65.4	76.5	70.2	74.1	83.8	81.2	-	-	-
UVLTrack-L [56]	-	-	-	64.8	-	68.8	71.3	-	78.3	51.2	-	59.0
SUTrack-L224 [14]	-	-	-	66.7	-	70.3	73.5	83.3	80.9	54.0	65.3	61.7
SUTrack-L384 [14]	-	-	-	67.9	-	72.1	75.2	84.9	83.2	53.6	64.2	60.5
ATCTrack-L	74.0	86.5	76.1	68.6	85.8	75.0	74.7	87.1	82.3	55.4	66.8	64.0

Table 1. Comparison with state-of-the-art trackers on four popular benchmarks: MGIT [33], TNL2K [76], LaSOT [19], and LaSOT$_{ext}$ [20]. The best two results are highlighted in red and blue, respectively.

Visual memory guidance. For the m^t that incorporates both current and historical target-context feature information, we adopt parameter-free attention operations [80] to facilitate its interaction with the search feature f_{XL}^t:

$$f_R^t = softmax(\frac{f_{XL}^t \cdot (m^t)^T}{\sqrt{D}}) \cdot m^t, \tag{8}$$

Here, $f_R^t \in \mathbb{R}^{N_x \times D}$ denotes the search feature that incorporates both textual and visual memory cues, which is subsequently fed into the prediction head.

3.4. Memory Storage Module

As previously mentioned, this module provides the tracker (at t) with stored memory features M^{t-1} (up to $t-1$). Meanwhile, it stores the newly generated memory unit m^t for guiding the tracking process at the next time step $(t+1)$. The L_m memory units in M are initialized with $f_{[C]}^0$ from the first frame and updated using a simple yet widely adopted sliding window approach [7, 80]. For further details, please refer to Appendix C.2.

3.5. Prediction Head and Loss

Based on the search feature f_R^t which integrates multi-modal cues, we utilize a classic CNN-based prediction head [80, 86] to obtain the final bounding box. We employ the focal loss L_{cls} [41], L_1 loss, and the generalized IoU loss L_{iou} [63] to supervise the prediction of bounding box, which are widely used in tracker design[86]. Additionally, for the target words classification task, we employ the binary cross-entropy loss L_{bce} for supervision. The overall loss function is formulated as follows:

$$L_{all} = L_{cls} + 2 \times L_{iou} + 5 \times L_1 + L_2 + 0.2 \times L_{bce}. \tag{9}$$

4. Experiments

4.1. Implementation Details

We adopt RoBERTa-Base [52] as our text encoder and, following recent advanced trackers [14, 66, 80], employ HiViT [69, 90] as our vision encoder. To balance performance and computational efficiency, we develop two model variants: ATCTrack-B and ATCTrack-L, initialized with Fast-iTPN-B and Fast-iTPN-L [69] respectively, with token dimensions D of 512 and 768. The template patches and search images are sized at 128×128 and 256×256, respectively. Additionally, the default length of MSM is set to four, and the dynamic template update strategy follows the STARK [82]. Our tracker is trained on a server with four A5000 GPUs and tested on an RTX-3090 GPU. The tracking speed of ATCTrack-B/L is 35/30 FPS. For a comparison of our model with mainstream VLTs in terms of parameters and speed, please refer to Appendix F.1.

We train our model using the training splits from La-

SOT [19], TNL2K [76], RefCOCOg [57], OTB99-Lang [49], VastTrack [61], GOT-10k [36], and TrackingNet [60]. For GOT-10k [36] and TrackingNet [60] that lack text annotations, we follow the All-in-One's strategy [87] by treating class names as pseudo language labels. Each training sample comprises a text description, two template patches, and four search frames from the same video sequence. Our tracker performs iterative training by sequentially selecting search images. The network parameters are optimized using AdamW optimizer [53] for 150 epochs, with each epoch containing 20,000 randomly sampled instances.

4.2. Comparison with State-of-the-arts

MGIT. MGIT is a latest VLT benchmark specifically tailored for the complex long-term scenarios. Each sequence contains challenging spatio-temporal causal relationships and is annotated with text prompts at three levels of granularity [26, 33, 34, 45–47]. As shown in Tab. 1, ATCTrack demonstrates superior performance at the representative action granularity. Particularly, ATCTrack-B surpassing the SOTA tracker MemVLT [25] by 4.3%, 3.2%, and 6.4% in area under the curve (AUC), normalized precision (P_{Norm}), and precision score (P), respectively. Unlike MemVLT's implicit modulation of static multimodal cues, ATCTrack explicitly adapts these cues from a target-context perspective. These results validate our approach's effectiveness in handling complex long-term scenarios.

TNL2K. TNL2K is also designed for the VLT task, and the introduction of attributes such as "adversarial samples" and "modality switch" significantly adds to the challenges [76]. As shown in Tab. 1, ATCTrack-L outperforms the recent ChatTracker-L by 4.8% in P. Compared to ChatTracker employs multimodal large language models to generate high-quality text annotations for addressing static text limitations, our textual target-context guidance module achieves superior performance with fewer network parameters.

LaSOT & LaSOT$_{ext}$. They are extensions of traditional visual tracking benchmarks [19, 79] by adding text labels, focusing on long-term tracking challenges. Furthermore, LaSOT$_{ext}$ also includes many similar distractors, further complicating the tracking task. As shown in Tab. 1, except for marginally lower AUC and P scores compared to SUTrack-L384, which utilizes larger image resolution and more training datasets, ATCTrack demonstrates highly competitive performance across all other metrics. The outstanding performance across multiple benchmarks further reflects ATCTrack's strong generalization capability.

Qualitative comparison. As shown in Fig. 4, we present the tracking results of ATCTrack-B and two existing SOTA VLTs[25, 56] on four challenging sequences. In these cases, the appearance of the target undergoes significant changes [27, 51], and the language descriptions contain context words that may lead to ambiguity. It is evident that our ATCTrack exhibits greater robustness [11] and effectiveness. More cases can be found in Appendix G.

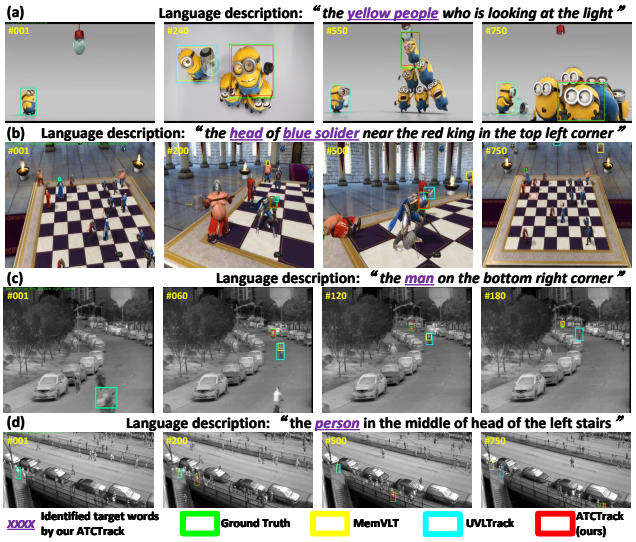

Figure 4. Qualitative comparison results of our tracker with other two VLTs (*i.e.*, MemVLT and UVLTrack) on four challenging cases. Better viewed in color with zoom-in.

#	Setting		TNL2K		LaSOT	
	$Textual_{TC}$	$Visual_{TC}$	AUC	P	AUC	P
1			65.5	70.6	72.5	79.8
2	✓		66.5	72.0	73.6	81.3
3		✓	66.7	72.3	73.7	81.7
4	✓	✓	67.5	73.6	74.7	82.3

Table 2. Ablation study on important model components.

4.3. Ablation Study

To investigate the properties of the various modules in ATCTrack, we conduct comprehensive ablation studies on ATCTrack-B using TNL2K [76] and LaSOT [19]. For the implementation details of each setting, see Appendix E.

Study on important model components. The core contribution of our work is the introduction of a novel multimodal target-context modeling mechanism. In Tab. 2 (#1), we show results using only the initial prompts and the dynamic template to guide tracker. Tab. 2 (#2) and (#3) present results using our textual and visual target-context guidance modules, respectively. Comparatively, Our method shows superior performance with 1.0% and 1.2% AUC gains on TNL2K, respectively, demonstrating the effectiveness of our approach. Additionally, Tab. 2 (#4) indicates that combining these two modules provides complementary benefits, further enhancing tracking performance.

Study on textual target-context modeling. Similar to most VLTs [85, 94, 98], Tab. 3 (#1) (identical to Tab. 2 (#1))

#	Setting	TNL2K		LaSOT	
		AUC	P	AUC	P
1	*naive method*	65.5	70.6	72.5	79.8
2	*+ target words awareness*	65.9	71.7	72.9	80.4
3	*+ context words calibration*	66.5	72.0	73.6	81.3
4	*- dual-type textual guidance*	66.2	71.8	73.2	80.3

Table 3. Ablation study on textual target-context modeling.

#	Setting	TNL2K		LaSOT	
		AUC	P	AUC	P
1	*naive method*	65.5	70.6	72.5	79.8
2	*w RoI*	65.6	71.1	71.1	77.4
3	*w search + crop mask*	65.9	71.4	71.9	78.7
4	*w search + global mask*	66.7	72.3	73.7	81.7

Table 4. Ablation study on visual target-context modeling.

#	Setting	TNL2K		LaSOT	
		AUC	P	AUC	P
1	*ATCTrack-B*	67.5	73.6	74.7	82.3
2	*w/o HiViT backbone*	65.8 (↓ 1.7)	71.3 (↓ 2.3)	72.9 (↓ 1.8)	80.7 (↓ 1.6)
3	*w/o dynamic template*	67.2 (↓ 0.3)	73.0 (↓ 0.6)	73.2 (↓ 1.5)	80.8 (↓ 1.5)
4	*w/o Textual$_{TC}$&Visual$_{TC}$*	65.5 (↓ 2.0)	70.6 (↓ 3.0)	72.5 (↓ 2.2)	79.8 (↓ 2.5)
5	*w/o target words label*	67.0 (↓ 0.5)	72.9 (↓ 0.7)	73.5 (↓ 1.2)	80.6 (↓ 1.7)

Table 5. Ablation study on the contribution of different modules.

treats the text features as a whole, indiscriminately fusing them with visual features. Tab. 3 (#2) and (#3) successively incorporate the target words awareness and context words calibration mechanisms to model and leverage the textual target-context cues aligned with the target state. The progressive improvement in performance demonstrates the effectiveness of our proposed textual guidance approach.

Besides, we utilize both the initial and calibrated textual feature f_{LC} for tracking guidance. Tab. 3 (#4) presents the results of using only the calibrated textual feature $f_{L'}$. Compared to Tab. 3 (#3), the degraded model performance indicates that dual-type textual features help the tracker leverage more comprehensive information.

Moreover, we quantitatively evaluate the target words identification accuracy of our method compared to the existing vision-text similarity-based method (see Appendix B for details). As shown in Fig. 2 (a), despite using only a lightweight multilayer perceptron (Eq. (1)), our method achieves impressive accuracy, distinguishing target words with an overall accuracy (Acc_{all}) of 98.9 %. This facilitates the tracker's utilization of textual target-context cues. Furthermore, our experiments reveal that lightweight text analysis tools, such as Scene Graph Parse [64], demonstrate poor performance in target word recognition, achieving an accuracy ($\text{Acc}_{\text{target}}$) of only 21.0%.

Study on visual target-context modeling. Tab. 4 (#1) shows the results when only sparse initial and dynamic template patches are employed. To model and leverage denser dynamic target-context features, we propose our visual target-context guidance module. Tab. 4 (#3, #4, #5) explores different implementation approaches. Tab. 4 (#3) adopts RoI processing [73], a classic local image feature modeling method, which replaces $h^t \odot f_X^t$ in Eq. (6) with the local region in f_X^t determined by the predicted bbox ($\times 1.5$). Similarly, Tab. 4 (#4) replaces h^t with the local mask of predicted bbox ($\times 1.5$). Compared to Tab. 4 (#2), these two methods achieve slight improvements on TNL2K but suffer significant performance drops on LaSOT. In contrast, our method (*i.e.*, Tab. 4 (#5)) achieves superior results, demonstrating the necessity of representing target-context information from a global perspective.

Further ablation studies. Several components in our model, such as the HiViT backbone [69, 90] and dynamic template [82], are well-established designs for improving tracking performance. Therefore, it is necessary to assess their impact relative to our core contribution, *i.e.*, the multimodal target-context guidance modules. As shown in Tab. 5 (#2, #3), replacing HiViT with ViT [86] or removing the dynamic template leads to performance degradation, which aligns with existing findings[80, 82]. However, Tab. 5 (#4) reveals that removing our multimodal target-context guidance module results in more significant performance drops, indicating that ATCTrack's superior performance primarily stems from our proposed method.

Furthermore, ATCTrack employs unique target word supervision labels, which constitute a significant contribution of our method. To ensure fairness, Tab. 5 (#5) presents the results without this label. Although the performance of the model decreases, it still surpasses the SOTA trackers represented by MemVLT [25] and SUTrack-B224 [14].

5. Conclusion

In pursuit of robust vision-language tracking, especially in complex long-term scenarios that reflect real-world conditions as exemplified by MGIT, we propose a novel tracker named ATCTrack. Through comprehensive target-context modeling, we obtain multimodal cues aligned with dynamic target states, offering an innovative solution to the obstacle where initial static cues fail to provide sustained guidance. For the textual modality, we introduce a precise target words awareness method to ensure sufficient attention to target words, and design an innovative context words calibration mechanism to mitigate potential misleading effects. For the visual modality, we efficiently characterize temporal target-context features to provide timely visual cues for tracking. Through these combined efforts, our model achieves outstanding performance, significantly surpassing existing methods across four mainstream benchmarks.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (Grant No.62176255).

References

[1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 3:11–12, 2022. 3

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 2, 4

[3] Yifan Bai, Zeyang Zhao, Yihong Gong, and Xing Wei. Artrackv2: Prompting autoregressive tracker where to look and how to describe. *arXiv preprint arXiv:2312.17133*, 2023. 7

[4] Moshe Bar. Visual objects in context. *Nature Reviews Neuroscience*, 5(8):617–629, 2004. 4

[5] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 7

[6] Wenrui Cai, Qingjie Liu, and Yunhong Wang. Learning historical status prompt for accurate and robust visual tracking. *arXiv preprint arXiv:2311.02072*, 2023. 7

[7] Wenrui Cai, Qingjie Liu, and Yunhong Wang. Hiptrack: Visual tracking with historical prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19258–19267, 2024. 6, 3

[8] Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. Robust object modeling for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9589–9600, 2023. 7

[9] Monica S Castelhano and Chelsea Heaven. The relative contribution of scene context and target features to visual search in scenes. *Attention, Perception, & Psychophysics*, 72(5): 1283–1297, 2010. 4

[10] 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. In *European Conference on Computer Vision*, pages 375–392. Springer, 2022. 7

[11] Honghao Chen, Yurong Zhang, Xiaokun Feng, Xiangxiang Chu, and Kaiqi Huang. Revealing the dark secrets of extremely large kernel convnets on robustness. *arXiv preprint arXiv:2407.08972*, 2024. 7

[12] 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. 7

[13] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14572–14581, 2023. 7

[14] Xin Chen, Ben Kang, Wanting Geng, Jiawen Zhu, Yi Liu, Dong Wang, and Huchuan Lu. Sutrack: Towards simple and unified single object tracking. *arXiv preprint arXiv:2412.19138*, 2024. 6, 8, 5, 7

[15] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, pages 6668–6677, 2020. 7

[16] 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. 7, 8

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3, 4, 5

[18] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 5

[19] 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 5369–5378, 2019. 2, 4, 6, 7, 1, 5

[20] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, et al. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129: 439–461, 2021. 2, 6, 1, 5, 7

[21] Qi Feng, Vitaly Ablavsky, Qinxun Bai, and Stan Sclaroff. Robust visual object tracking with natural language region proposal network. *arXiv preprint arXiv:1912.02048*, 1(7):8, 2019. 5

[22] 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. 5

[23] 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 5847–5856, 2021. 3, 5

[24] Xiaokun Feng, Dailing Zhang, Shiyu Hu, Xuchen Li, Meiqi Wu, Jing Zhang, Xiaotang Chen, and Kaiqi Huang. Enhancing vision-language tracking by effectively converting textual cues into visual cues. *arXiv preprint arXiv:2412.19648*, 2024. 6, 5

[25] Xiaokun Feng, Xuchen Li, Shiyu Hu, Dailing Zhang, Jing Zhang, Xiaotang Chen, Kaiqi Huang, et al. Memvlt: Vision-language tracking with adaptive memory-based prompts. *Advances in Neural Information Processing Systems*, 37: 14903–14933, 2025. 3, 4, 5, 6, 7, 8

[26] Xiaokun Feng, Haiming Yu, Meiqi Wu, Shiyu Hu, Jintao Chen, Chen Zhu, Jiahong Wu, Xiangxiang Chu, and Kaiqi Huang. Narrlv: Towards a comprehensive narrative-centric evaluation for long video generation models. *arXiv preprint arXiv:2507.11245*, 2025. 7

[27] Xiaokun Feng, Dailing Zhang, Shiyu Hu, Xuchen Li, Meiqi Wu, Jing Zhang, Xiaotang Chen, and Kaiqi Huang. Cstrack: Enhancing rgb-x tracking via compact spatiotemporal features. *arXiv preprint arXiv:2505.19434*, 2025. 7

[28] 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. 7

[29] Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18686–18695, 2023. 7

[30] Mingzhe Guo, Zhipeng Zhang, Heng Fan, and Liping Jing. Divert more attention to vision-language tracking. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 4446–4460, 2022. 2, 5

[31] Mingzhe Guo, Zhipeng Zhang, Heng Fan, Liping Jing, Yilin Lyu, Bing Li, and Weiming Hu. Learning target-aware representation for visual tracking via informative interactions. *arXiv preprint arXiv:2201.02526*, 2022. 7

[32] Lingyi Hong, Shilin Yan, Renrui Zhang, Wanyun Li, Xinyu Zhou, Pinxue Guo, Kaixun Jiang, Yiting Chen, Jinglun Li, Zhaoyu Chen, et al. Onetracker: Unifying visual object tracking with foundation models and efficient tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19079–19091, 2024. 3, 6, 5, 7

[33] Shiyu Hu, Dailing Zhang, Meiqi Wu, Xiaokun Feng, Xuchen Li, Xin Zhao, and Kaiqi Huang. A multi-modal global instance tracking benchmark (mgit): Better locating target in complex spatio-temporal and causal relationship. In *the 37th Conference on Neural Information Processing Systems*, pages 25007–25030, 2023. 1, 2, 4, 6, 7, 5

[34] Shiyu Hu, Xin Zhao, Lianghua Huang, and Kaiqi Huang. Global instance tracking: Locating target more like humans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):576–592, 2023. 7

[35] Shiyu Hu, Xin Zhao, and Kaiqi Huang. Sotverse: A user-defined task space of single object tracking. *International Journal of Computer Vision*, 132:872–930, 2024. 2

[36] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1562–1577, 2019. 7

[37] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1

[38] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 5

[39] Yutong Kou, Jin Gao, Bing Li, Gang Wang, Weiming Hu, Yizheng Wang, and Liang Li. Zoomtrack: Target-aware non-uniform resizing for efficient visual tracking. *Advances in Neural Information Processing Systems*, 36:50959–50977, 2023. 7

[40] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015. 1

[41] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 6

[42] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 3

[43] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, pages 4282–4291, 2019. 7

[44] Xuchen Li, Xiaokun Feng, Shiyu Hu, Meiqi Wu, Dailing Zhang, Jing Zhang, and Kaiqi Huang. Dtllm-vlt: Diverse text generation for visual language tracking based on llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7283–7292, 2024. 2

[45] Xuchen Li, Shiyu Hu, Xiaokun Feng, Dailing Zhang, Meiqi Wu, Jing Zhang, and Kaiqi Huang. Dtvlt: A multi-modal diverse text benchmark for visual language tracking based on llm. *arXiv preprint arXiv:2410.02492*, 2024. 7

[46] Xuchen Li, Shiyu Hu, Xiaokun Feng, Dailing Zhang, Meiqi Wu, Jing Zhang, and Kaiqi Huang. How texts help? a fine-grained evaluation to reveal the role of language in vision-language tracking. *arXiv preprint arXiv:2411.15600*, 2024.

[47] Xuchen Li, Shiyu Hu, Xiaokun Feng, Dailing Zhang, Meiqi Wu, Jing Zhang, and Kaiqi Huang. Visual language tracking with multi-modal interaction: A robust benchmark. *arXiv preprint arXiv:2409.08887*, 2024. 7

[48] Yihao Li, Jun Yu, Zhongpeng Cai, and Yuwen Pan. Cross-modal target retrieval for tracking by natural language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4931–4940, 2022. 2

[49] Zhenyang Li, Ran Tao, Efstratios Gavves, Cees GM Snoek, and Arnold WM Smeulders. Tracking by natural language specification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6495–6503, 2017. 1, 2, 3, 4, 7

[50] Liting Lin, Heng Fan, Zhipeng Zhang, Yaowei Wang, Yong Xu, and Haibin Ling. Tracking meets lora: Faster training, larger model, stronger performance. In *ECCV*, 2024. 7

[51] Xinran Ling, Chen Zhu, Meiqi Wu, Hangyu Li, Xiaokun Feng, Cundian Yang, Aiming Hao, Jiashu Zhu, Jiahong Wu, and Xiangxiang Chu. Vmbench: A benchmark for perception-aligned video motion generation. *arXiv preprint arXiv:2503.10076*, 2025. 7

[52] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 4, 6

[53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7, 4

[54] Ding Ma and Xiangqian Wu. Tracking by natural language specification with long short-term context decoupling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14012–14021, 2023. 6, 5

[55] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoniu Song, Xing Chen, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. *arXiv preprint arXiv:2502.10248*, 2025. 5

[56] Yinchao Ma, Yuyang Tang, Wenfei Yang, Tianzhu Zhang, Jinpeng Zhang, and Mengxue Kang. Unifying visual and vision-language tracking via contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4107–4116, 2024. 3, 6, 7, 5

[57] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 7, 1

[58] Zhongjie Mao, Yucheng Wang, Xi Chen, and Jia Yan. Textual tokens classification for multi-modal alignment in vision-language tracking. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8025–8029. IEEE, 2024. 2, 4, 1, 3, 5

[59] Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *ICCV*, pages 13444–13454, 2021. 7

[60] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European conference on computer vision (ECCV)*, pages 300–317, 2018. 7

[61] Liang Peng, Junyuan Gao, Xinran Liu, Weihong Li, Shaohua Dong, Zhipeng Zhang, Heng Fan, and Libo Zhang. Vasttrack: Vast category visual object tracking. *Advances in Neural Information Processing Systems*, 37:130797–130818, 2025. 7, 1

[62] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 3, 5

[63] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019. 6

[64] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015. 8, 3

[65] Yanyan Shao, Shuting He, Qi Ye, Yuchao Feng, Wenhan Luo, and Jiming Chen. Context-aware integration of language and visual references for natural language tracking. *arXiv preprint arXiv:2403.19975*, 2024. 2, 3, 4, 6, 1, 5

[66] Liangtao Shi, Bineng Zhong, Qihua Liang, Ning Li, Shengping Zhang, and Xianxian Li. Explicit visual prompts for visual object tracking. *arXiv preprint arXiv:2401.03142*, 2024. 6, 7

[67] Yiming Sun, Fan Yu, Shaoxiang Chen, Yu Zhang, Junwei Huang, Yang Li, Chenhui Li, and Changbo Wang. Chattracker: Enhancing visual tracking performance via chatting with multimodal large language model. *Advances in Neural Information Processing Systems*, 37:39303–39324, 2025. 6, 5

[68] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020. 3

[69] Yunjie Tian, Lingxi Xie, Jihao Qiu, Jianbin Jiao, Yaowei Wang, Qi Tian, and Qixiang Ye. Fast-itpn: Integrally pre-trained transformer pyramid network with token migration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 6, 8

[70] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 4, 1

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proceedings of the Advances in Neural Information Processing Systems*, 30, 2017. 5, 4

[72] Hongyu Wang, Xiaotao Liu, Yifan Li, Meng Sun, Dian Yuan, and Jing Liu. Temporal adaptive rgbt tracking with modality prompt. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5436–5444, 2024. 3

[73] 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. 3, 8, 5

[74] Rong Wang, Zongheng Tang, Qianli Zhou, Xiaoqian Liu, Tianrui Hui, Quange Tan, and Si Liu. Unified transformer with isomorphic branches for natural language tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 4, 5

[75] Xiao Wang, Chenglong Li, Rui Yang, Tianzhu Zhang, Jin Tang, and Bin Luo. Describe and attend to track: Learning natural language guided structural representation and visual attention for object tracking. *arXiv preprint arXiv:1811.10014*, 2018. 5

[76] 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. 2, 4, 6, 7, 1, 5

[77] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9697–9706, 2023. 7

[78] Qiangqiang Wu, Tianyu Yang, Ziquan Liu, Baoyuan Wu, Ying Shan, and Antoni B Chan. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14561–14571, 2023. 7

[79] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(09):1834–1848, 2015. 7

[80] Jinxia Xie, Bineng Zhong, Zhiyi Mo, Shengping Zhang, Liangtao Shi, Shuxiang Song, and Rongrong Ji. Autoregressive queries for adaptive tracking with spatio-temporaltransformers. *arXiv preprint arXiv:2403.10574*, 2024. 6, 8, 3, 7

[81] Chenlong Xu, Bineng Zhong, Qihua Liang, Yaozong Zheng, Guorong Li, and Shuxiang Song. Less is more: Token context-aware learning for object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8824–8832, 2025. 2

[82] 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. 5, 6, 8, 4, 7

[83] 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. 3

[84] Dawei Yang, Jianfeng He, Yinchao Ma, Qianjin Yu, and Tianzhu Zhang. Foreground-background distribution modeling transformer for visual object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10117–10127, 2023. 7

[85] Zhengyuan Yang, Tushar Kumar, Tianlang Chen, Jingsong Su, and Jiebo Luo. Grounding-tracking-integration. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(9):3433–3443, 2021. 3, 7, 5

[86] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *Proceedings of the European Conference on Computer Vision*, pages 341–357, 2022. 3, 5, 6, 8, 7

[87] Chunhui Zhang, Xin Sun, Yiqian Yang, Li Liu, Qiong Liu, Xi Zhou, and Yanfeng Wang. All in one: Exploring unified vision-language tracking with multi-modal alignment. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5552–5561, 2023. 3, 6, 7, 5

[88] Dailing Zhang, Shiyu Hu, Xiaokun Feng, Xuchen Li, Jing Zhang, Kaiqi Huang, et al. Beyond accuracy: Tracking more like human via visual search. *Advances in Neural Information Processing Systems*, 37:2629–2662, 2025. 3, 5

[89] Guangtong Zhang, Bineng Zhong, Qihua Liang, Zhiyi Mo, Ning Li, and Shuxiang Song. One-stream stepwise decreasing for vision-language tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 3, 1, 5

[90] Xiaosong Zhang, Yunjie Tian, Lingxi Xie, Wei Huang, Qi Dai, Qixiang Ye, and Qi Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In *The Eleventh International Conference on Learning Representations*, 2022. 6, 8

[91] Haojie Zhao, Xiao Wang, Dong Wang, Huchuan Lu, and Xiang Ruan. Transformer vision-language tracking via proxy token guided cross-modal fusion. *Pattern Recognition Letters*, 168:10–16, 2023. 5

[92] Xin Zhao, Shiyu Hu, Yipei Wang, Jing Zhang, Yimin Hu, Rongshuai Liu, Haibin Ling, Yin Li, Renshu Li, Kun Liu, and Jiadong Li. Biodrone: A bionic drone-based single object tracking benchmark for robust vision. *International Journal of Computer Vision*, 132:1659–1684, 2024. 2

[93] Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhenjun Tang, Rongrong Ji, and Xianxian Li. Leveraging local and global cues for visual tracking via parallel interaction network. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(4):1671–1683, 2022. 2

[94] Yaozong Zheng, Bineng Zhong, Qihua Liang, Guorong Li, Rongrong Ji, and Xianxian Li. Towards unified token learning for vision-language tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 3, 5, 6, 7, 4, 8

[95] Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. Odtrack: Online dense temporal token learning for visual tracking. *arXiv preprint arXiv:2401.01686*, 2024. 3, 7

[96] Yaozong Zheng, Bineng Zhong, Qihua Liang, Ning Li, and Shuxiang Song. Decoupled spatio-temporal consistency learning for self-supervised tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10635–10643, 2025. 3

[97] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 3

[98] Li Zhou, Zikun Zhou, Kaige Mao, and Zhenyu He. Joint visual grounding and tracking with natural language specification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23151–23160, 2023. 2, 3, 6, 7, 1, 4, 5, 8

ATCTrack: Aligning Target-Context Cues with Dynamic Target States for Robust Vision-Language Tracking

Supplementary Material

A. Target Words Annotation Pipeline

Given the inherently flexible and diverse nature of textual descriptions, it is challenging for trackers to accurately identify target words and context words. In our work, we approach the identification of target words as a multi-label binary classification task, enhancing the model's ability to recognize target words through supervised learning. However, existing benchmarks [20, 33, 61, 76] provide only textual descriptions without labeled information on the types of target words (*i.e.*, target words or context words). For such a natural language processing task, we leverage the powerful text understanding capabilities of the large language models [37, 70] to construct an automated target words annotation pipeline. Specifically, we employ the widely-used multimodal large language model, GPT-4o [37], and have devised a specific core prompt to guide GPT-4o in recognizing target words (as shown in Fig. A1).

Leveraging our automated annotation pipeline, we complete the labeling of target words in textual data from the MGIT [33], TNL2K [76], LaSOT [19], RefCOCOg [57], OTB99-Lang [49] and Vasttrack [61] datasets. We conduct a random sampling of the labeled results, inspect 50 sentences, and find that the annotations are entirely accurate. This ensures the reliability of our supervised models in classifying target words. In the future, we will open source both the target words label information and our code.

B. Evaluation of Target Words Identification

In this section, we discuss the specific implementation methods for the target words classification accuracy results shown in Fig. 2 (a). Recent studies, such as QueryNLT [65], TTCTrack [58] and OSDT [89], have utilized vision-text similarity metrics to identify target words. Although this is one of their main contributions, they have not provided quantitative evaluation results. For this, we conduct a quantitative analysis based on the target words label information obtained from Sec. A.

B.1. Similarity-Based Target Words Identification

Considering that QueryNLT [65], TTCTrack [58] and OSDT [89] have not open-sourced their code, we employ JointNLT [98], a representative vision-language tracker, as a proxy model for evaluation. The core insight of JointNLT is the use of a one-stream network to jointly model the feature extraction and interaction of text, template images, and search images. The extensive feature interaction among these elements can, to some extent, represent the feature interaction operations conducted in the aforementioned works for measuring vision-text similarity.

Specifically, at time step $t(t \geq 0)$, after the feature encoding by the JointNLT's backbone network, we obtain the visual features $f_V^t \in \mathbb{R}^{400\times512}$ and the textual features $f_L^t \in \mathbb{R}^{L\times512}$. Here, the length of the visual tokens is fixed at 400, while the length of the textual tokens, L, is determined by the number of words in the sentence. The similarity between them is obtained through the following operations:

$$att_{vl}^t = (f_L^t)^T \cdot f_V^t, \tag{A1}$$

where $att_{vl}^t \in \mathbb{R}^{L\times400}$ represents the similarity between each visual and textual token. By averaging along the dimension of the search tokens, we can determine the attention each textual token receives at the current time step t, denoted as $att_l^t \in \mathbb{R}^L$.

By concatenating att_l^t at each time step in a video sequence along the time dimension, we can obtain a heatmap of textual feature information for this sequence, denoted as $Att_l \in \mathbb{R}^{L\times T}$, where T represents the number of frames in the video sequence.

For a more intuitive understanding, we conduct a visualization analysis using two video sequences as examples. The related results are depicted in Fig. A2, which serves as a supplement to Fig. 2 (b) and (c) in the main text. As shown in Fig. A2 (a), the target being tracked in this sequence is "plane". In the corresponding Att_l heatmap, the target word "plane" receives significant attention, indicating that the tracker correctly understands the intent embedded in the text prompt, and this text cue aids in the tracking process. For the example in Fig. A2 (b), the intended tracking target is "yellow people", but the tracker primarily focuses on the word "the light". This indicates that the tracker did not correctly focus on the target words, which could mislead the tracking process.

B.2. Evaluation of Target Words Identification Accuracy

In addition to qualitatively demonstrating the tracker's ability to distinguish each word in the text as described above, we also need to conduct a quantitative evaluation. First, to analyze the tracker's attention to each word throughout the entire video sequence, we average Att_l along the time dimension, resulting in $Res_l \in \mathbb{R}^L$. Each element in Res_l reflects the amount of attention the tracker gives to the word at the corresponding position.

You are an expert in linguistic analysis for dynamic visual tracking. Your task is to analyze a text description of a target object in a video and identify which phrases describe the **target's intrinsic attributes** (stable properties that remain consistent with the object's physical essence) vs. **contextual attributes** (dynamic properties that may change with scene evolution). Finally, output the phrases of the target's intrinsic attributes in a structured format.

Rules:
1. **Target's intrinsic attributes** must satisfy:
- Directly describe the target's inherent physical properties (e.g., category, color, material, shape, brand)
- Remain valid even if the target changes pose, location, or interacts with other objects
- Examples: "red", "car", "striped", "round glasses"

2. **Contextual attributes** must be:
- Related to the target's temporary state or environment (e.g., position, motion, relative relationships)
- Likely to become invalid due to scene dynamics
- Examples: "on the left", "jumping", "next to a chair"

Examples:
1. Input: "a white van parked beside a traffic light"
Output: [{"phrase": "white", "reason": "color is a stable property"},
{"phrase": "van", "reason": "object category"}]

2. Input: "the running black cat with a collar"
Output: [{"phrase": "black", "reason": "color attribute"},
{"phrase": "cat", "reason": "object category"},
{"phrase": "collar", "reason": "physical accessory"}]

3. Input: "the second man from left to right direction"
Output: [{"phrase": "man", "reason": "object category"}]

Question: The input is {xx}, what should the corresponding output be ?

Figure A1. **Prompt used to guide GPT-4o in identifying target words information.** This prompt primarily consists of two parts: task requirement descriptions and example guidance. Replace {xx} with the sentence to be identified to achieve output results similar to the example format.

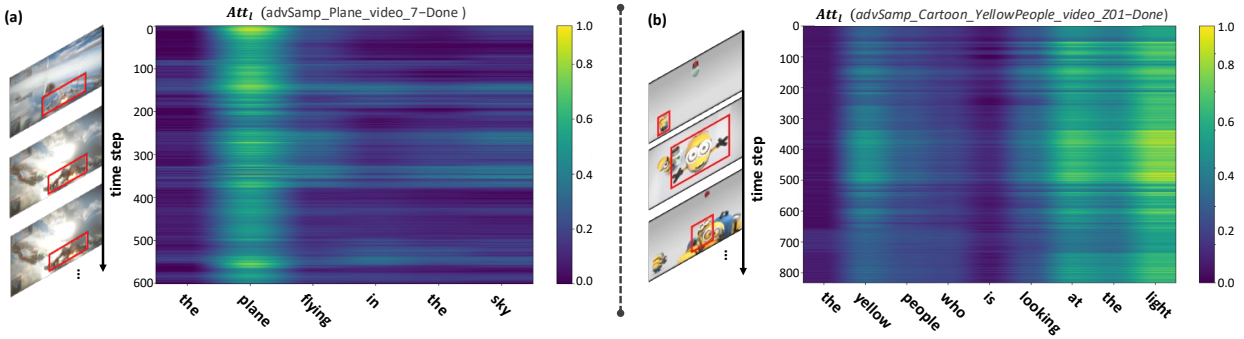

Figure A2. **Visualization results of Att_l across two video sequences.** **(a)** In the sequence 'advSamp_Plane_video_7-Done', the target "plane" receives significant attention during the tracking process, which aligns with our intended effect. **(b)** In the sequence 'advSamp_Cartoon_YellowPeople_video_Z01-Done', the target "yellow people" is intended to be tracked, but the tracker primarily focuses on the text "the light". This indicates that the tracker did not correctly focus on the target words, which could mislead the tracking process.

Based on this information, we can map to obtain the tracker's final prediction results for target words classification, $p \in \{0,1\}^L$. Specifically, based on the target words label information obtained from Sec. A, we can determine the number of target words k in the sentence. Then, we calculate the top k elements and their indices in Res_l. Sub-

sequently, we set the elements at these indices in p to 1, while all other elements are set to 0.

Additionally, utilizing the target words label information provided in Sec. A, we can obtain a ground truth label $g \in \{0,1\}^L$. In this label, 0 indicates that the word token at that position is a context word, and 1 indicates it is a target word. We then establish two accuracy assessment metrics, namely, Acc_{all} and $\text{Acc}_{\text{target}}$, by performing different calculations on p and g to evaluate the tracker's accuracy in classifying target words. Here, Acc_{all} represents the overall classification accuracy of the model for both target and context words; while $\text{Acc}_{\text{target}}$ focuses on the classification accuracy specifically for target words.

$$\text{Acc}_{\text{all}} = \frac{\sum_{i=1}^{L} \mathbf{1}(p_i = g_i)}{L}, \tag{A2}$$

$$\text{Acc}_{\text{target}} = \frac{\sum_{i=1}^{N} \mathbf{1}(p_i = 1 \wedge g_i = 1)}{\sum_{i=1}^{N} \mathbf{1}(g_i = 1)}. \tag{A3}$$

Here, $\mathbf{1}(\cdot)$ is an indicator function that returns 1 if the condition within the parentheses is satisfied.

Similarly, for our proposed ATCTrack and its predictions about target words p_{lt} (see Eq. (1)), we can use the same method to map it to p, and then use the above formula for accuracy measurement. The corresponding accuracy results are displayed in Fig. 2 (a). It is evident that our method significantly outperforms methods based on vision-text similarity in both metrics.

B.3. Analysis of Evaluation Results

Fig. 2 (a) shows the target words identification accuracy of our method compared to the existing vision-text similarity-based method [58, 65, 89]. As can be seen, our method achieves an impressive 96.7% in the $\text{Acc}_{\text{target}}$ metric, significantly surpassing the latter's 29.9%. This precise target word awareness lays a solid foundation for subsequent text cue adjustment and utilization. This demonstrates that our lightweight multilayer perceptron (Eq. (1)) effectively transfers the LLMs' target word distinguishing capability into the tracker. Although existing LLMs have good target word sensing capabilities, integrating LLMs directly into the tracker incurs substantial computational costs, which is detrimental to practical applications. Additionally, there are lightweight text component analysis tools in the field of natural language processing, such as the widely used Scene Graph Parser [64]. We evaluated the Scene Graph Parser's accuracy in identifying target words in sentences and found it to be only 21.0%. This indicates that these tools are not yet capable of meeting our target word identification needs in a plug-and-play manner.

C. More Details on the ATCTrack

Due to space constraints, we focus primarily on the main contributions of our paper in the Sec. 3, specifically the textual target-context guidance module (see Sec. 3.2) and the visual target-context guidance module (see Sec. 3.3). For other components of our tracker, such as the prediction head and memory storage module, we provide a brief introduction using current mainstream methods, supplemented by relevant references. In this section, we offer an additional explanation of these components.

C.1. Prediction Head

The prediction head is used to predict the final bbox b^t. We employ a CNN-based tracking head [80, 86], which is widely adopted in tracker design. Firstly, for the search feature $f_R^t \in \mathbb{R}^{N_x \times D}$ that integrates both textual and visual cues, we transform it into a 2D spatial feature map. Subsequently, after passing through L_h stacked Conv-BN-ReLU layers, we obtain a classification score map $P \in [0,1]^{1 \times H_s \times W_s}$, the size of the bbox $B \in [0,1]^{2 \times H_s \times W_s}$, and the offset size $O \in [0,1)^{2 \times H_s \times W_s}$. Then, the position with the highest classification score is considered to be the target position, *i.e.*, $(x_d, y_d) = \arg\max_{(x,y)} P_{xy}$. The final target bbox is obtained as:

$$x = x_d + O(0, x_d, y_d), \tag{A4}$$

$$y = y_d + O(1, x_d, y_d), \tag{A5}$$

$$w = S(0, x_d, y_d), \tag{A6}$$

$$h = S(1, x_d, y_d). \tag{A7}$$

C.2. Memory Storage Module

As introduced in Sec. 3.4, we employ the sliding windows method [7, 80] to update memory units, a method widely used in recent vision trackers focused on temporal modeling. The visual memory feature M in MSM consists of a list of L_m memory units m, denoted as $M = \{m_i\}_{i=1}^{L_m}$. Below, we will illustrate how the sliding windows memory storage method is implemented.

For a video sequence with T frames ($0 \leq t \leq T-1$), the memory units in M need to be initialized when processing the first frame (*i.e.*, $t = 0$). Specifically, after encoding the visual input information via a vision encoder, we obtain the feature $f_{[C]}^0$ encoded from the [CLS] token. Considering that the [CLS] token can represent global visual features [17], we use $f_{[C]}^0$ to initialize the L_m memory units. During the time interval $t \in [1, T-1]$, after tracking each search frame, we obtain the updated memory unit m^t. We pop the memory unit with index 0 from M and append m^t to the end of M.

Model	Params	Speed	AUC	P
JointNLT [98]	153M	31FPS	56.9	58.1
MMTrack [94]	177M	37FPS	58.6	59.4
MemVLT [25]	175M	32FPS	63.3	67.4
ATCTrack-B	160M	35FPS	67.5	73.6
ATCTrack-L	340M	30FPS	68.6	75.0

Table A1. Results of efficiency analysis.

D. More Details on Model Implementation

Due to space constraints, only core model implementation details are provided in Sec. 4.1. Here, we supplement some additional details. First, regarding the model structure, when performing context words calibration, we use two stacked modules consisting of Eq. (3) and Eq. (4). When executing visual memory representation, we use two stacked modules consisting of Equations Eq. (6) and Eq. (7). It is important to note that we only use the FFN in the visual memory representation part. Considering that the computational cost of FFN in Transformer modules is higher than that of Attention [71], our module design helps reduce the overall parameters and computation of the model.

Additionally, for model training, we use the AdamW optimizer [53] to optimize our model. The text encoder remains frozen, the learning rate is set to 10^{-5} for the vision encoder, 10^{-4} for the remaining unfrozen modules, and the weight decay is set to 10^{-4} . We train for a total of 150 epochs and reduce the learning rate by a factor of 10 after 120 epochs. Finally, during the model inference stage, dynamic template updating follows the implementation of STARK [82]. We set the update interval to 25 and the update confidence threshold to 0.8.

E. Experimental Details of Ablation Studies

In Sec. 4.3, we conduct detailed ablation analyses to investigate the properties of the various modules in ATCTrack. Due to space limitations, we do not fully elaborate on the specific implementation of the ablation experiments. In this section, we provide additional details.

E.1. Ablation Study on important model components

Tab. 2 presents the ablation study results of two core components in our approach: the textual and the visual target-context guidance modules. The specific implementations are as follows:

Tab. 2 (#1) demonstrates the baseline results without our textual and visual object-context guidance modules. In this setup, textual features are processed as a whole entity, an approach widely adopted by recent trackers such as SNLT [74] and MMTrack [94]. Specifically, we employ a transformer-based decoder to facilitate interaction between textual features f_L and search features f_X^t:

$$f_R^t = Trans_{Dec}(f_X^t, f_L), \tag{A8}$$

where $Trans_{Dec}$ represents the standard transformer decoder layer [71], primarily consisting of attention operations and feed-forward networks. f_R^t denotes the search features embedded with textual cues, which are subsequently fed into the prediction head to obtain final tracking results. To ensure fair comparison, we configure the transformer decoder with four layers, matching the parameter count with the visual and textual object-context guidance module.

Tab. 2 (#2) shows the results using only the textual object-context guidance module. In this implementation, we omit the visual memory guidance process and directly feed the output features f_{XL}^t from the textual target-context guidance module into the prediction head to obtain final results.

Tab. 2 (#3) presents the results using only our visual object-context guidance module. In this implementation, we employ a transformer-based decoder to guide the search features with textual information, which is formulated as:

$$f_{XL}^t = Trans_{Dec}(f_X^t, f_L), \tag{A9}$$

For fair comparison, we implement a two-layer decoder architecture.

Tab. 2 (#4) demonstrates the results of our complete ATCtrack model.

E.2. Ablation Study on Textual Target-Context Modeling

Tab. 3 shows different ways of utilizing textual cues, with the specific implementations for each setting as follows:

Naive method. This setting is consistent with that of Tab. 2 (#1).

+ Target words awareness. This refers to the incorporation of target words awareness method based on the "*naive method*" setting. Specifically, we concatenate the f_{LT} with f_L to obtain context features f_{LC} for subsequent textual guidance.

+ Context words calibration. This refers to the incorporation of context words calibration operations based on the "*+ target words awareness*" setting. This is the approach adopted by our ATCTrack.

- Dual-type textual guidance. This approach utilizes only the calibrated single-type text features $f_{L'}$ for textual guidance, where $f_{LC} = f_{L'}$.

Method	MGIT (Action)			TNL2K			LaSOT			LaSOT_{ext}		
	AUC	P_{Norm}	P	AUC	P_{Norm}	P	AUC	P_{Norm}	P	AUC	P_{Norm}	P
Basic Variants												
Wang [75]	-	-	-	-	-	-	27.7	-	30.4	-	-	-
Feng [21]	-	-	-	25.0	34.0	27.0	50.0	-	56.0	-	-	-
Feng [22]	-	-	-	25.0	33.0	27.0	35.0	-	35.0	-	-	-
GTI [85]	-	-	-	-	-	-	47.8	-	47.6	-	-	-
TNL2K-II [76]	-	-	-	42.0	50.0	42.0	51.3	-	55.4	-	-	-
SNLT [23]	3.6	22.6	0.4	-	-	-	54.0	63.6	57.4	-	-	-
VLT_{TT} [30]	46.8	60.2	31.8	54.7	71.8	55.3	67.3	80.2	71.5	48.4	59.9	54.3
TransVLT [91]	-	-	-	56.0	61.7	-	66.4	-	70.8	-	-	-
JointNLT [98]	61.0	78.6	44.5	56.9	73.6	58.1	60.4	69.4	63.6	-	-	-
TransNLT [74]	-	-	-	57.0	75.0	57.0	60.0	-	63.0	-	-	-
DecoupleTNL [54]	-	-	-	56.7	-	56.0	71.2	-	75.3	-	-	-
All-in-One [87]	-	-	-	55.3	-	57.2	71.7	82.4	78.5	54.5	63.5	-
MMTrack [94]	-	-	-	58.6	75.2	59.4	70.0	82.3	75.7	49.4	59.9	55.3
QueryNLT [65]	-	-	-	56.9	73.6	58.1	59.9	69.6	63.5	-	-	-
TTCTrack [58]	-	-	-	58.1	-	-	67.6	-	-	48.8	-	-
OSDT [89]	-	-	-	59.3	76.2	61.5	64.3	73.4	68.6	-	-	-
OneTracker [32]	-	-	-	58.0	-	59.1	70.5	79.9	76.5	-	-	-
UVLTrack-B [56]	-	-	-	62.7	-	65.4	69.4	-	74.9	49.2	-	55.8
CTVLT [24]	69.2	-	62.9	62.2	-	79.5	72.3	-	79.7	-	-	-
ChatTracker-B [67]	-	-	-	59.6	76.3	62.1	71.7	80.9	77.5	-	-	-
MemVLT [25]	69.4	81.3	63.7	63.3	80.9	67.4	72.9	85.7	80.5	52.1	63.3	59.8
SUTrack-B224 [14]	-	-	-	65.0	-	67.9	73.2	83.4	80.5	53.1	64.2	60.5
SUTrack-B384 [14]	-	-	-	65.6	-	69.3	74.4	83.9	81.9	52.9	63.6	60.1
ATCTrack-B	73.7	84.5	70.1	67.5	85.3	73.6	74.6	87.0	82.1	54.6	65.7	62.8
Performance-oriented Variants												
ChatTracker-L [67]	-	-	-	65.4	76.5	70.2	74.1	83.8	81.2	-	-	-
UVLTrack-L [56]	-	-	-	64.8	-	68.8	71.3	-	78.3	51.2	-	59.0
SUTrack-L224 [14]	-	-	-	66.7	-	70.3	73.5	83.3	80.9	54.0	65.3	61.7
SUTrack-L384 [14]	-	-	-	67.9	-	72.1	75.2	84.9	83.2	53.6	64.2	60.5
ATCTrack-L	74.0	86.5	76.1	68.6	85.8	75.0	74.7	87.1	82.3	55.4	66.8	64.0

Table A2. Comparison with state-of-the-art vison-language trackers on four popular benchmarks: MGIT [33], TNL2K [76], LaSOT [19], and LaSOT_{ext} [20]. The best two results are highlighted in red and blue, respectively.

E.3. Ablation Study on Visual Target-Context Modeling

Tab. 4 shows different ways of utilizing visual cues, with the specific implementations for each setting as follows:

Naive method. This setting is consistent with that of Tab. 2 (#1).

+ ROI. This represents the augmentation of the "*naive method*" by incorporating explicit visual memory features for tracking assistance. Specifically, we employ the Region of Interest (RoI) approach [62], which is widely adopted in recent Visual-Language Trackers (VLTs) such as JointNLT [98] and TrDiMP [73]. We apply RoI processing to the search features f_X^t using the predicted bounding box scaled by 1.5 to obtain localized search features $f_{X'}^t \in \mathbb{R}^{36\times D}$. Subsequently, the visual memory representation process is implemented through the following computations:

$$f_{[C]M'} = Norm(f_{[C]M} + \Phi_{CA}(f_{[C]M}, f_{X'}^t)), \quad \text{(A10)}$$

$$f_{[C]M''} = Norm(f_{[C]M'} + FFN(f_{[C]M'})). \quad \text{(A11)}$$

+ Search + crop mask. This setting involves using a local mask to construct the object-context indication map. Specifically, for the global object-context indication map

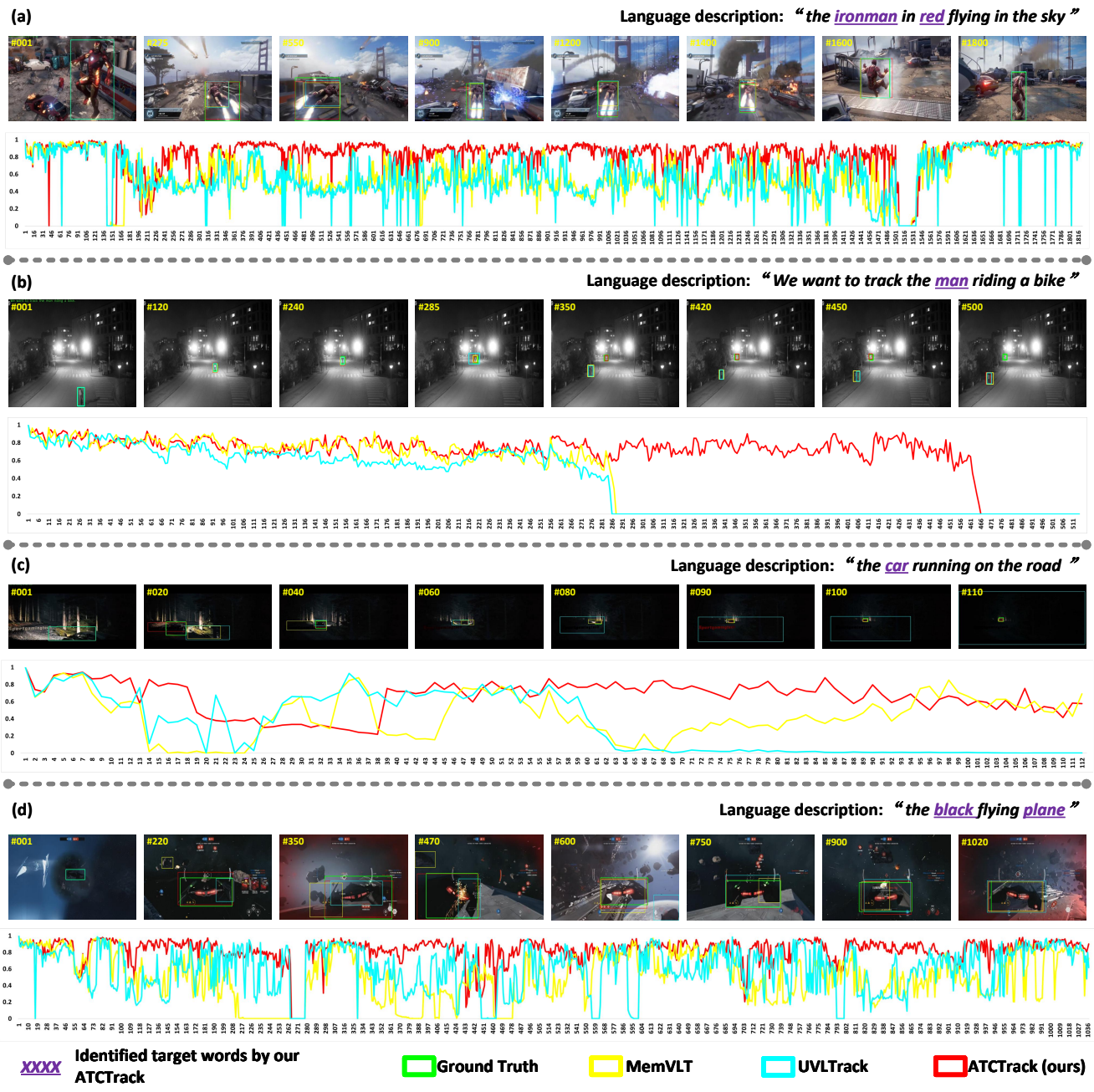

Figure A3. Qualitative comparison results of our tracker with other two state-of-the-art vision-language trackers (*i.e.*, MemVLT and UVLTrack) on four challenging cases. For each video case, we select representative frames to illustrate the predicted bounding boxes of each model and plot the curves of the IOU predictions across the entire video. Better viewed in color with zoom-in.

h^t, we retain only the values within the area corresponding to 1.5 times the predicted bbox, while setting the values in all other areas to zero, resulting in h_l^t. Then, the visual memory representation process is implemented through the following computations:

$$f_{[C]M'} = Norm(f_{[C]M} + \Phi_{CA}(f_{[C]M}, h_l^t \odot f_X^t)), \quad \text{(A12)}$$

$$f_{[C]M''} = Norm(f_{[C]M'} + FFN(f_{[C]M'})). \quad \text{(A13)}$$

+ Search + global mask. This setting involves using a global mask to construct the object-context indication map,

Method	TNL2K			LaSOT			LaSOT$_{ext}$		
	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P
Basic Variants									
SiamFC [5]	-	-	-	29.5	45.0	28.6	33.6	42.0	33.9
SiamRPN++ [43]	-	-	-	41.3	48.2	41.2	49.6	56.9	49.1
SiamBAN [15]	-	-	-	41.0	48.5	41.7	51.4	59.8	52.1
TransT [12]	-	-	-	64.9	73.8	69.0	-	-	-
Stark [82]	-	-	-	67.1	77.0	-	-	-	-
KeepTrack [59]	-	-	-	67.1	77.2	70.2	-	-	-
Mixformer [16]	-	-	-	69.2	78.7	74.7	-	-	-
TransInMo [31]	52.0	58.5	52.7	65.7	76.0	70.7	-	-	-
OSTrack-256 [86]	54.3	-	-	69.1	78.7	75.2	47.4	57.3	53.3
OSTrack-384 [86]	55.9	-	-	71.1	81.1	77.6	50.5	61.3	57.6
AiATrack [28]	-	-	-	69.0	79.4	73.8	47.7	55.6	55.4
SimTrack [10]	-	-	-	69.3	78.5	-	-	-	-
GRM [29]	-	-	-	69.9	79.3	75.8	-	-	-
SeqTrack-B256 [13]	54.9	-	-	69.9	79.7	76.3	49.5	60.8	56.3
SeqTrack-B384 [13]	56.4	-	-	71.5	81.1	77.8	50.5	61.6	57.5
ARTrack-256 [77]	57.5	-	-	70.4	79.5	76.6	46.4	56.5	52.3
ARTrack-384 [77]	59.8	-	-	72.6	81.7	79.1	51.9	62.0	58.5
OSTrack-Zoom [39]	56.5	-	57.3	70.2	-	76.2	50.5	-	57.4
DropTrack [78]	56.9	-	57.9	71.8	81.8	78.1	52.7	63.9	60.2
ROMTrack-256 [8]	-	-	-	69.3	78.8	75.6	48.9	59.3	55.0
ROMTrack-384 [8]	-	-	-	71.4	81.4	78.2	51.3	62.4	58.6
F-BDMTrack-256 [84]	56.4	-	56.5	69.9	79.4	75.8	47.9	57.9	54.0
F-BDMTrack-384 [84]	57.8	-	59.4	72.0	81.5	77.7	50.8	61.3	57.8
EVPTrack-224 [66]	57.5	-	58.8	70.4	80.9	77.2	48.7	59.5	55.1
EVPTrack-384 [66]	59.1	-	62.0	72.7	82.9	80.3	53.7	65.5	61.9
ODTrack-B [95]	60.9	-	-	73.2	83.2	80.6	52.4	63.9	60.1
AQATrack-256 [80]	57.8	-	59.4	71.4	81.9	78.6	51.2	62.2	58.9
AQATrack-384 [80]	59.3	-	62.3	72.7	82.9	80.2	52.7	64.2	60.8
ARTrackV2-256 [3]	-	-	-	71.6	80.2	77.2	50.8	61.9	57.7
ARTrackV2-384 [3]	-	-	-	73.0	82.0	79.6	52.9	63.4	59.1
HIPTrack [6]	-	-	-	72.7	82.9	79.5	53.0	64.3	60.6
OneTracker [32]	58.0	-	59.1	70.5	79.9	76.5	-	-	-
LoRAT-B224 [50]	58.8	-	61.3	71.7	80.9	77.3	50.3	61.6	57.1
LoRAT-B378 [50]	59.9	-	63.7	72.9	81.9	79.1	53.1	64.8	60.6
SUTrack-B224 [14]	65.0	-	67.9	73.2	83.4	80.5	53.1	64.2	60.5
SUTrack-B384 [14]	65.6	-	69.3	74.4	83.9	81.9	52.9	63.6	60.1
ATCTrack-B	67.5	85.3	73.6	74.6	87.0	82.1	54.6	65.7	62.8
Performance-oriented Variants									
ODTrack-L [95]	61.7	-	-	74.0	84.2	82.3	53.9	65.4	61.7
LoRAT-L224 [50]	61.1	-	65.1	74.2	83.6	80.9	52.8	64.7	60.0
LoRAT-L378 [50]	62.3	-	67.0	75.1	84.1	82.0	56.6	69.0	65.1
SUTrack-L224 [14]	66.7	-	70.3	73.5	83.3	80.9	54.0	65.3	61.7
SUTrack-L384 [14]	67.9	-	72.1	75.2	84.9	83.2	53.6	64.2	60.5
ACTrack-L	68.6	85.8	75.0	74.7	87.1	82.3	55.4	66.8	64.0

Table A3. Comparison with state-of-the-art vision-only trackers on three popular benchmarks: TNL2K [76], LaSOT [19], and LaSOT$_{ext}$ [20]. The best two results are highlighted in red and blue, respectively.

which is used to obtain explicit visual memory features. This is the approach adopted by our ATCTrack.

E.4. Ablation Study on the Contribution of different modules

w/o HiViT backbone. This setting refers to replacing the HiViT backbone [69, 90] with the ViT backbone typically used in conventional trackers [16, 86].

w/o dynamic template. This setting refers to using only the original static template for visual input, without the sparse dynamic template [82].

w/o Textual$_{TC}$ & Visual$_{TC}$. This setting is the same as setting in Tab. 2 (#1), meaning that the visual and textual target-context guidance mechanism we designed is not utilized.

w/o target words label. This setting, with the model structure unchanged, refers to not using target words supervision signals, thus excluding L_{bce} loss.

F. Additional Experimental Results

F.1. Efficiency Analysis

In Tab. A1, we compare ATCTrack with the latest VLTs (*i.e.*, JointNLT [98], MMTrack [94], and MemVLT [25]) in terms of efficiency (Params and Speed) and performance (AUC and P on TNL2K). For ATCTrack-B, the parameters and tracking speed are comparable to recent trackers, but it shows significant performance advantages, such as a 4.2% improvement in AUC compared to MemVLT. For ATCTrack-L, the parameter scale is considerably larger than ATCTrack-B, which leads to a further performance improvement.

F.2. Comparison with More Trackers

In Tab. 1 of Sec. 4.2, due to space constraints, we compare ATCTrack with several recent high-performance vision-language trackers. As a supplement, Tab. A2 presents the performance of a broader range of vision-language trackers. Additionally, in line with the prevailing paradigm of vision-language tracking models [25, 94, 98], Tab. A3 provides additional comparisons with vision-only trackers. The strong performance of our model among these trackers further demonstrates the effectiveness of our approach.

G. More Qualitative Results

Due to space limitations, Fig. 4 only presents four cases for the qualitative comparison between our model and the latest SOTA models. In this section, we provide additional qualitative comparison results, as illustrated in Fig. A3.