# Learning Frequency and Memory-aware Prompts for Multi-modal Object Tracking

Boyue Xu[a], Ruichao Hou[a], Tongwei Ren[a], Dongming Zhou[b], Gangshan Wu[a], Jinde Cao[c,d]

[a]*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210008, Jiangsu, China*
[b]*School of Information Science and Engineering, Yunnan University, Kunming, 650091, Yunnan, China*
[c]*School of Mathematics, Southeast University, Nanjing, 210096, Jiangsu, China*
[d]*Purple Mountain Laboratories, Nanjing, 211111, Jiangsu, China*

**Abstract**

Prompt-learning-based multi-modal trackers inject auxiliary-modality cues into frozen foundation models via lightweight adapters. Yet most methods fuse modalities only in the spatial or channel domain, ignoring frequency-domain discrepancies that can amplify cross-modal noise, and they rely on short-term temporal cues from adjacent frames, making tracking prone to drift under occlusion and long-term appearance changes. To address these issues, we present learning frequency and memory-aware prompts, a dual-adapter framework that injects lightweight prompts into a frozen RGB tracker. A frequency-guided visual adapter adaptively transfers complementary cues across modalities by jointly calibrating spatial, channel, and frequency components, narrowing the modality gap without full fine-tuning. A multilevel memory adapter with short, long, and permanent memory stores, updates, and retrieves reliable temporal context, enabling consistent propagation across frames and robust recovery from occlusion, motion blur, and illumination changes. The unified design preserves the efficiency of prompt learning while strengthening cross-modal interaction and temporal coherence. Extensive experiments on RGB-Thermal, RGB-Depth, and RGB-Event benchmarks show consistent state-of-the-art results over fully fine-tuned and adapter-based baselines. Code and models are available at https://github.com/xuboyue1999/mmtrack.git.

*Keywords:* Multi-modal tracking, prompt learning, frequency-guided fusion, memory mechanism, temporal modeling.

## 1. Introduction

Visual object tracking (VOT) [1] aims to localize the target annotated in the first frame throughout subsequent frames [2] and underpins a wide range of applications in autonomous driving [3], embodied artificial intelligence [4], and human–computer interaction [5]. Despite the significant progress of RGB-only trackers, they can still drift or miss the target in complex scenarios due to the inherent limitations of visible-spectrum sensors, such as background clutter and low illumination. To enhance robustness, recent works introduce auxiliary modalities and develop multi-modal tracking tasks, including RGB-Thermal (RGB-T) [6], RGB-Depth (RGB-D) [7], and RGB-Event (RGB-E) [8]. Multi-modal tracking methods generally fall into two paradigms.

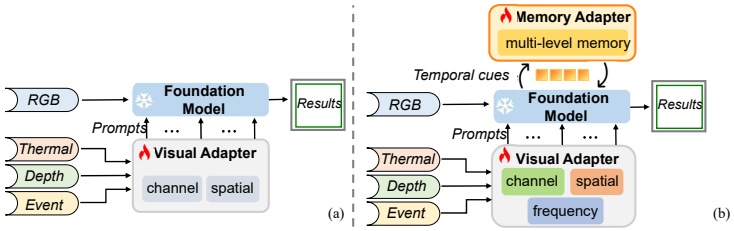


Figure 1: Framework comparisons between the existing prompt-learning-based tracker and our tracker. (a) Existing trackers propagate temporal cues from adjacent frames and fuse multi-modal features in channel and spatial dimensions. (b) The proposed method integrates a memory adapter to propagate cues adaptively and merge features in channel, spatial, and frequency dimensions.

The first is the classic dual-branch architecture [9, 10, 11, 12], typically tailored to a specific multi-modal setting and trained end-to-end with full fine-tuning. While effective at capturing modality-specific representations, such methods are constrained by limited training data and high computational cost. The second paradigm adopts prompt learning [13, 14, 15], which fine-tunes lightweight adapters on frozen RGB-based backbones, improving adaptability and training efficiency. Despite notable progress, two critical gaps remain. (1) Frequency cues are essential in multi-modal tracking [16, 17], yet current prompt-learning trackers underuse frequency-domain information and thus struggle to bridge the modality gap. Different modalities exhibit complementary frequency characteristics due to their sensing principles, as shown in Fig. 2. RGB en-

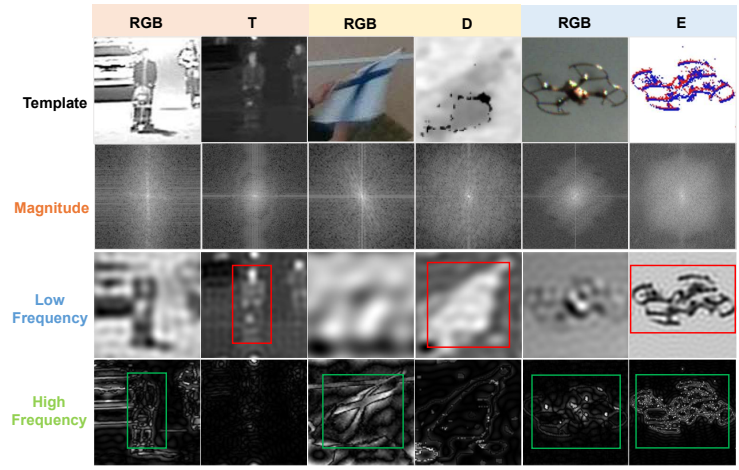


Figure 2: Illustration of frequency-domain characteristics for RGB-T, RGB-D, and RGB-E. The second to fourth rows show the magnitude map, the low-frequency visualization, and the high-frequency visualization, respectively. Red boxes indicate the most informative regions in the low-frequency domain, while green boxes highlight the most informative regions in the high-frequency domain.

codes fine-grained high-frequency textures, thermal and depth emphasize low-frequency structural contours, and event streams carry sparse, sharp high-frequency motion edges. Direct spatial-domain fusion is therefore suboptimal owing to cross-modality noise and misaligned frequency content. Although recent efforts explore frequency-aware designs [18, 19], they largely rely on hand-crafted frequency separation, limiting data-driven exploitation. (2) Temporal cues are equally crucial, yet existing trackers [20, 21, 22] typically restrict modeling to adjacent-frame propagation or confidence-based template updates, which fail to capture long-range dependencies, errors accumulate under occlusion, fast motion, or appearance changes, degrading robustness.

To address these two core challenges, we propose a unified multi-modal tracker that learns frequency and memory-aware prompts via a dual-adapter architecture comprising a frequency-guided visual adapter and a multi-level memory adapter (Fig. 1(b)). For effective cross-modal fusion, the visual adapter introduces a frequency selector that adaptively emphasizes informative subbands to refine intra-modal features. Specifically, it extracts high-frequency textures from RGB and low-frequency structures from auxiliary modalities. By selectively fusing these compatible cues, our method miti-

gates the interference caused by intrinsic feature discrepancies, thereby narrowing the modality gap without full fine-tuning.

To model temporal dependencies, the memory adapter is inspired by the human memory mechanism [23, 24]. It maintains a multi-level memory pool to store global temporal cues, together with update and retrieval operations that refresh the memory per frame and select reliable cues for subsequent tracking. The design learns memory-aware prompts that propagate consistent temporal context across frames and recover from occlusion, fast motion, and illumination changes. Extensive experiments on mainstream multi-modal tracking benchmarks demonstrate that our tracker outperforms fully fine-tuned and prompt-learning baselines in both accuracy and robustness, while preserving the efficiency of adapter-based tuning.

In summary, our main contributions are as follows:

• We present a unified dual-adapter framework that learns frequency and memory-aware prompts for multi-modal tracking, showing consistent gains on RGB-T, RGB-D, and RGB-E tasks.

• We develop a lightweight frequency-guided visual adapter that aggregates informative cues across frequency, spatial, and channel dimensions to produce modality-aware prompts and enhance cross-modal fusion.

• We propose a multi-level memory adapter that stores and retrieves global temporal cues with update and retrieval operations, enabling adaptive propagation of temporal context along video sequences.

## 2. Related Work

### *2.1. Multi-modal Object Tracking*

Multi-modal object tracking incorporates additional modality with the RGB modality, such as thermal, depth, or event data, to enhance the perceptual capabilities of visible sensors, particularly in complex scenarios where visual may be invalid [25].

Most multi-modal trackers are typically extensions of strong RGB-based trackers, encompassing multi-modal feature extraction and fusion to strengthen representations. Previous works follow the two-parallel branch architecture. For example, APFNet [26]

explored fusion strategies under various challenging attributes to boost tracking accuracy. MTNet [9] leveraged Transformers to establish the global association for multi-modal feature interaction and reinforcement. Likewise, SPT [7] applied Transformers both in feature extraction and fusion, maximizing the utilization of complementary multi-modal data. However, these methods relying on parallel feature extraction structures for both modalities, can introduce considerable computational overhead and training complexity, complicating cross-modal transfer processes. More recently, prompt learning-based methods have emerged, aiming to develop lightweight adapters on the powerful foundation models to fine-tune them, thereby reducing training costs and enhancing scalability [14]. For instance, ViPT [14] proposed an adapter based on prompt learning, enabling efficient multi-modal fusion in tracking tasks. SDSTrack [13] introduced a data augmentation for lower-quality modalities, improving tracking performance on specific challenging attributes. OneTracker [15] further expanded the input of multi-modal trackers by incorporating text prompts. SUTrack [27] designs a unified tracking framework capable of handling a wide range of single-modal and multi-modal tracking tasks.

Despite the remarkable achievements, existing methods often ignore the importance of frequency information, limiting their performance. This motivates us to propose a novel visual adapter that fully extracts multi-modal cues from multiple dimensions, with a particular emphasis on selecting high- and low-frequency characteristics.

### *2.2. Temporal Modeling in Visual Tracking*

It is well recognized that rich temporal information is essential for visual object tracking, and effectively enhancing context propagation in the temporal domain remains a central focus of current research.

One type of method focuses on designing template update strategies to replace the initial target template over time. For instance, Stark [28] combined the initial template features with online information to achieve an adaptive template update. Yang et al. [21] introduced a multi-frame template pool to select the optimal template, mitigating the unreliability of single-frame templates. SDSTrack [13] applied the template updated mechanism in the tracking framework by using confidence scores to choose ap-

propriate templates. While these methods improve tracking robustness to some extent, they still treat tracking as a frame-by-frame template-matching task without leveraging deeper temporal correlations. Another type of method [29] attempts to propagate temporal context across frames. For instance, TCTrack [20] propagated template cues between adjacent frames to guide more precise template feature extraction. ODTrack [30] incorporated global tokens into the attention mechanism, improving temporal propagation efficiency. ASTMT [22] applied a propagation network in infrared tracking, enhancing temporal transmission. SeqTrack [31], by contrast, feeds the entire sequence into the tracker and performs tracking from a global perspective.

While these advanced methods underscore the value of continuous temporal information in tracking, relying solely on adjacent-frame propagation risks being misled by noisy or erroneous data. Unlike existing trackers, our proposed memory adapter investigates global tracking cues, adaptively propagating the temporal relationships among successive frames for more robust tracking.

### *2.3. Efficient Single-modal Tracking.*

Balancing tracking accuracy with computational efficiency is a long-standing pursuit in the RGB tracking. Pioneering works have explored designing compact architectures to achieve high-speed inference. For instance, LightTrack [32] employs neural architecture search to automatically synthesize lightweight backbones and heads with low FLOPs. E.T.Track [33] introduces an exemplar transformer to replace expensive cross-correlation operations, while HiT [34] incorporates a bridge module to adapt lightweight hierarchical transformers for efficient tracking. Similarly, FEAR [35] explores dual-template updating for efficient robust tracking.

While these methods provide valuable insights into efficient architecture design, they are inherently tailored for single-modal scenarios. Directly extending these specific architectures to multi-modal tracking often necessitates duplicating backbones or designing complex fusion branches, which inevitably compromises their lightweight design. We employ parameter-efficient prompt learning. This allows us to inject multi-modal cues into a frozen foundation model, maintaining high efficiency without redesigning the backbone architecture.

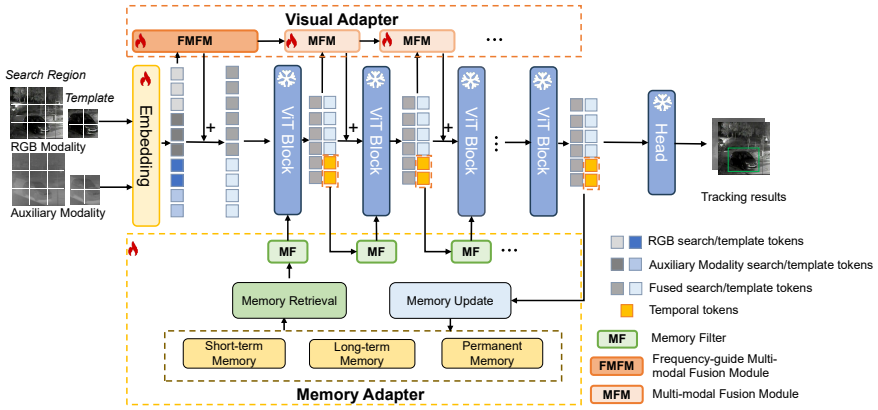


Figure 3: The framework of the proposed method. We first transform the templates and search region of each modality into tokens, then concatenate them with temporal cue tokens and feed them into the $L$-layer ViT block. The visual adapter and memory adapter are parallel with the ViT block. The memory adapter is used to propagate the valuable temporal cues across frames, and the visual adapter is used for modality interaction and fusion. The output features are fed into the prediction head to produce the tracking results.

## 3. Method

### 3.1. Preliminary and Notation

**Problem Formulation.** Given a pair of multi-modal sequence and an initial bounding box, the multi-modal tracking task can learn a tracker $T : \{\mathbf{X}^t_{rgb}, \mathbf{X}^t_x, \mathbf{Z}^0_{rgb}, \mathbf{Z}^0_x\} \rightarrow \mathbf{B}^t$, where $\mathbf{X}^t_{rgb}, \mathbf{X}^t_x$ represent the $t$-th search frames of RGB and auxiliary modality (*e.g.,* thermal, depth, or event), $\mathbf{Z}^0_{rgb}, \mathbf{Z}^0_x$ are the template of RGB and auxiliary modality generated by the initial bounding box $\mathbf{B}^0$, $\mathbf{B}^t$ represents the $t$-th predicted bounding box.

To fully harness the potential of prompt learning, we propose a visual adapter $V$ and integrate the multi-template mechanism [30] into multi-modal tracking. Additionally, to effectively capture and propagate temporal tracking cues, we propose a memory adapter $M$. The overall tracking process can be described as follows:

$$\begin{aligned} M :&\mathbf{C}^{t-1} \rightarrow \mathcal{U}\{M\}, \\ &\mathcal{R}\{M\} \rightarrow \mathbf{C}^t, \end{aligned} \tag{1}$$

$$T : \{V\{\mathbf{X}^t_{rgb}, \mathbf{X}^t_x\}, V\{\mathbf{Z}^0_{rgb}, \mathbf{Z}^0_x, ..., \mathbf{Z}^i_{rgb}, \mathbf{Z}^i_x\}, \mathbf{C}^t\} \rightarrow \mathbf{B}^t, \tag{2}$$

where $\mathbf{C}^t$ is the $t$-th temporal tracking cue, $\mathcal{U}\{\cdot\}$ and $\mathcal{R}\{\cdot\}$ represent memory update and retrieval operation, respectively. $\mathbf{Z}_{rgb}^t$ and $\mathbf{Z}_x^t$ denote the $t$-th templates generated by the corresponding frames and tracking result.

**Foundation Model.** We choose a powerful RGB tracker ODTrack [30] as the foundation model. Given the input search region and template $\mathbf{X}_{rgb}$ and $\mathbf{Z}_{rgb}$, they are first sent into patch embedding layer to obtain 1D tokens $\mathcal{H}_x^0$ and $\mathcal{H}_z^0$. These tokens are then concatenated to form the input token $\mathcal{H}^0 = [\mathcal{H}_x^0, \mathcal{H}_z^0]$. The input tokens are fed into $L$-layer ViT block encoder, and the output is passed through a box head to generate the tracking results. The propagation process can be formulated as follows:

$$\begin{aligned} \mathcal{H}^l &= \mathcal{E}^l(\mathcal{H}^{l-1}), l = 1, 2, 3, ..., L, \\ \mathbf{B} &= \phi(\mathcal{H}^L), \end{aligned} \tag{3}$$

where $\mathcal{E}(\cdot)$ is the ViT block and $\phi(\cdot)$ is prediction head.

### *3.2. Overall Framework*

The proposed framework is illustrated in Figure 3. It consists of four main components: the ViT backbone, visual adapter, memory adapter, and prediction head. Initially, the templates and search regions for both RGB and auxiliary modalities are embedded into tokens through the patch-embedding layer. These tokens are then sent to the frequency-guided multi-modal fusion module (FMFM) for shallow feature fusion. Subsequently, the temporal tracking cue tokens are retrieved from the multi-level memory pool, passed through a memory filter, and then sent to the ViT block along with the search region and template tokens. After each ViT block, the output undergoes the multi-modal fusion module (MFM) for modality enhancement and fusion, while the temporal tracking cues pass through the memory filter. After passing through $L$ layers of ViT blocks, the final tokens are used in the head operation to obtain the tracking results, and temporal tracking cues are stored in the multi-level memory pool.

### *3.3. Visual Adapter*

The visual adapter plays a crucial role in prompt-learning-based multi-modal tracking methods, as it directly influences how effectively multi-modal information is leveraged. Our objective is to design a visual adapter that fully explores the potential of

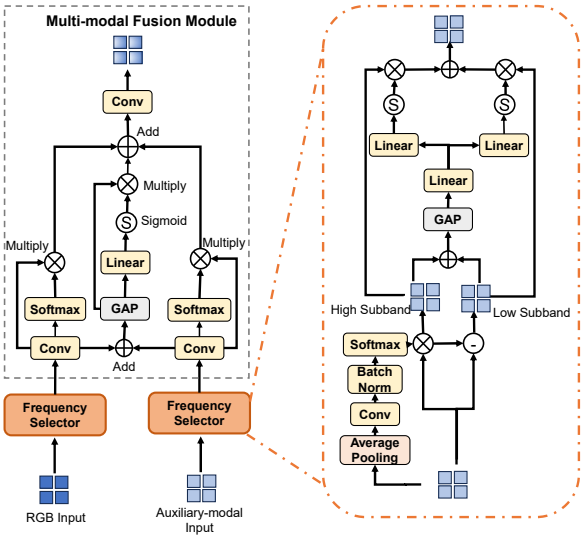


Figure 4: Detailed design of the frequency-guided multi-modal fusion module, which enhances the feature representation by combining spatial, channel, and frequency information from different modalities.

multi-modal data while maintaining efficiency. To achieve this, we propose a frequency-guided multi-modal fusion module for the first layer of the visual adapter, followed by multi-modal fusion modules for subsequent layers. With this design, we can extract frequency information in the shallow layer while adhering to the prompt-learning principle of maintaining parameter efficiency.

**Frequency Selector.** Frequency is a key attribute in image. To effectively leverage this information, the frequency selector operates through a two-stage process consisting of frequency decomposition and adaptive selection. First, the module performs frequency decomposition to isolate distinct feature components. Inspired by existing frequency separation methods [16, 17], we employ an average pooling operation to extract the low-frequency subband, which typically represents structural information. Mathematically, average pooling functions as a spatial low-pass filter that smooths out local high-frequency variations, effectively retaining the structural information. The high-frequency subband containing texture details is then derived by subtracting the low-frequency component from the original features. Second, the module executes adaptive selection via a weighted fusion mechanism. Rather than applying a discrete

hard selection, we utilize a learnable attention network to generate continuous weights for both frequency subbands. The soft selection process enables the network to emphasize informative frequencies while suppressing noise, resulting in a more robust fused representation. It is worth noting that unlike fixed frequency filters, our module is fully learnable. Through end-to-end training, the network automatically optimizes the decomposition filters and selection weights to adapt to the specific characteristics of different modalities, thereby achieving data-driven adaptive adjustment without manual tuning. The detailed design is shown in the right part of Figure 4. We first separate the input into high-frequency and low-frequency components, which can be calculated as follows:

$$\mathbf{F}_{high} = \mathbf{F}_{ori} \otimes (\text{Softmax}(\text{BN}(\text{Conv}(\text{Ap}(\mathbf{F}_{ori}))))), \tag{4}$$

$$\mathbf{F}_{low} = \mathbf{F}_{ori} - \mathbf{F}_{high}, \tag{5}$$

where $\text{Ap}(\cdot)$, $\text{Conv}(\cdot)$, and $\text{BN}(\cdot)$ represent average pooling, convolution, and batch normalization, respectively. $\mathbf{F}_{ori}$, $\mathbf{F}_{high}$, and $\mathbf{F}_{low}$ denote the input feature, the high- and low-frequency features, respectively. Then, we select and fuse the different frequency features to get the more representative features, which can be calculated as follows:

$$\mathbf{F}_{global} = \text{FC}\left(\text{GAP}\left(\mathbf{F}_{high} \oplus \mathbf{F}_{low}\right)\right), \tag{6}$$

$$\hat{\mathbf{F}_{high}} = \sigma\left(\text{FC}_{high}\left(\mathbf{F}_{global}\right)\right) \otimes \mathbf{F}_{high}, \tag{7}$$

$$\hat{\mathbf{F}_{low}} = \sigma\left(\text{FC}_{low}\left(\mathbf{F}_{global}\right)\right) \otimes \mathbf{F}_{low}, \tag{8}$$

where $\text{FC}(\cdot)$ is the linear layer, $\text{GAP}(\cdot)$ is global average pooling, $\sigma$ is Sigmoid, $\oplus$ denotes element-wise addition and $\otimes$ denotes the element-wise multiplication operation. Finally, we use element-wise addition to combine the high-frequency and low-frequency components of the image.

**Multi-modal Fusion Module.** The multi-modal fusion module integrates the multi-modal information from both spatial and channel perspectives. As shown in the left part of Figure 4, the input is divided into three branches: two branches are dedicated to enhancing multi-modal features from a spatial perspective, highlighting the most informative features, while the third branch concatenates the dual modalities and selects the

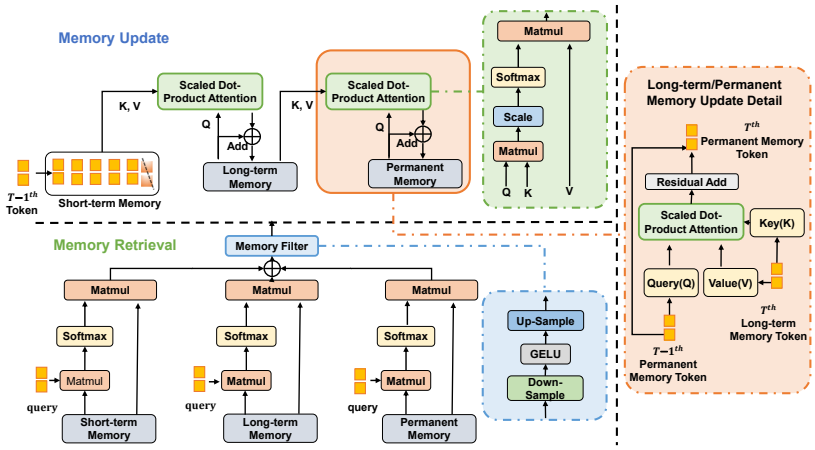


Figure 5: Detailed design of memory update and memory retrieval, which ensures the most reliable tracking cues are propagated in the subsequent sequence.

most relevant channels, effectively suppressing the influence of redundant information, which can be calculated as follows:

$$\mathbf{F}_i = \text{Conv}(\mathbf{I}_i), \quad i \in \{RGB, X\}, \tag{9}$$

$$\mathbf{F}_i^s = \mathbf{F}_i \otimes (\text{Softmax}(\mathbf{F}_i), \quad i \in \{RGB, X\}, \tag{10}$$

$$\mathbf{F}^g = \text{GAP}(\mathbf{F}_{RGB} \oplus \mathbf{F}_X), \tag{11}$$

$$\mathbf{F}^c = \mathbf{F}^g \otimes (\sigma(\text{FC}(\mathbf{F}^g)), \tag{12}$$

where $\mathbf{F}^s$, $\mathbf{F}^c$ represent the features passed through the spatial and channel branches, respectively. $\mathbf{I}_{RGB}$ and $\mathbf{I}_X$ are the inputs of RGB and auxiliary modality from the frequency selector in the first layer or those from the ViT backbone in subsequent layers. We then apply element-wise addition to combine the outputs of the three branches, followed by a convolution layer to produce the final output. This output is added to the original output from the previous ViT block and concatenated with the temporal tracking cue token, which serves as the input for the next ViT block.

We thoroughly exploit the potential of modality fusion from frequency, spatial, and channel perspectives, allowing us to fine-tune only a small number of parameters for prompt learning. The approach achieves impressive performance across a wide range of multi-modal tracking tasks.

### *3.4. Memory Adapter*

Memory mechanisms are commonly used in VOT to store temporal tracking cues [30, 20]. To further improve the robustness of multi-modal object tracking, we propose a memory adapter inspired by the human memory system, consisting of short-term memory, long-term memory, and permanent memory. The proposed multi-level memory operates through two key operations: memory update and memory retrieval. In the memory update operation, we use the previous tracking cue token to update all memory levels, while in the memory retrieval operation, the $T$-th temporal tracking cue is retrieved from the memory. Each level holds an $N \times H$ tensor, where $H = 768$ aligns with the ViT token dimension and $N$ is set to 8, 8, and 3 for short-term, long-term, and permanent memory, respectively. The short-term memory stores tokens from the most recent eight frames, while the long-term and permanent memory banks are hierarchically refreshed following a temporal update protocol.

The memory update and memory retrieval operations are shown in Figure 5. In the memory update operation, the initial cue tokens are stored in each level of memory at the start of tracking. After each frame during the tracking process, the temporal tracking cue token is first used to update the short-term memory, which stores the most recent 8 frames of temporal tracking cue tokens. Next, we employ a cross-attention mechanism to update the long-term memory, as detailed in the right part of Figure 5. Specifically, the existing long-term memory is the query ($\mathbf{Q}$) to actively select and absorb consistent features from the short-term memory (acting as key $\mathbf{K}$ and value $\mathbf{V}$), effectively filtering out transient noise. This process is calculated as follows:

$$\mathbf{LTM}^{'} = \text{Softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{K}^{T}}{\sqrt{d_k}}\right) \cdot \mathbf{V}, \tag{13}$$

$$\mathbf{LTM} = \mathbf{LTM} \oplus \mathbf{LTM}^{'}, \tag{14}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent the query, key, and value, respectively, the key and value are derived from the short-term memory, while the query comes from the long-term memory, $d_k$ represents the dimension of the key, the $\oplus$ represents element-wise addition. The update operation for the permanent memory follows the same procedure as that for the long-term memory.

In the memory retrieval operation, we aim to extract global tracking information from the memory while avoiding errors from incorrect memories. To achieve this, we use the latest temporal tracking cue as the query to select information from each level of memory. This selection operation can be calculated as follows:

$$\mathbf{W}_i = \text{Softmax}(\mathbf{q} \cdot \mathbf{M}_i^T), \quad i \in \{s, l, p\}, \tag{15}$$

$$\mathbf{C}_i = \mathbf{W}_i \otimes \mathbf{M}_i, \quad i \in \{s, l, p\}, \tag{16}$$

where Softmax(·) denotes the Softmax operation, $\mathbf{q}$ denotes the query. $\mathbf{W}_s$, $\mathbf{W}_l$, and $\mathbf{W}_p$ represent the weights of each memory cue in short-term memory, long-term memory, and permanent memory, respectively. Likewise, $\mathbf{M}_s$, $\mathbf{M}_l$, $\mathbf{M}_p$ refer to the stored memory in short-term, long-term, and permanent memory, while $\mathbf{C}_s$, $\mathbf{C}_l$, and $\mathbf{C}_p$ represent the selected memory from these levels, respectively. After selecting from each level of memory, we use element-wise addition to combine the results and then send them into the memory filter, which can be calculated as follows:

$$\mathbf{C} = \mathbf{C}_s \oplus \mathbf{C}_l \oplus \mathbf{C}_p, \tag{17}$$

$$\mathbf{C}^{'} = \text{Us}(\text{G}(\text{Ds}(\mathbf{C}))), \tag{18}$$

where Ds(·) and Us(·) represent the downsampling and upsampling operations, respectively. G(·) is the GELU activation function. The memory filter is applied after each ViT block to ensure that the temporal tracking cue is appropriately maintained and adjusted for each level of the ViT block.

### *3.5. Prediction Head and Loss Function*

We adopt the prediction head from the base tracker [30], freezing both the parameters of classification and regression heads. The classification head yields the score map, and the regression head generates the bounding box. These components work together to achieve the final tracking outcome.

The loss function contains classification loss and regression loss. The proposed method employs focal loss [36] as the classification loss $\mathcal{L}_{cls}$, which is suitable for the dataset with long-tail distribution and can be calculated as:

$$\mathcal{L}_{cls} = -\sum_{t} \alpha (1 - p_t)^{\gamma} log(p_t), \tag{19}$$

where $t$ represents the $t$-th samples, $\alpha_t$ is the weight coefficient, $p_t$ indicates the probability belonging to the foreground. The regression loss contains $\mathcal{L}_1$ loss and $GIoU$ loss, which can be calculated as:

$$\mathcal{L}_{reg} = \sum_{t} \left( \lambda_1 \mathcal{L}_1 \left( b_t, \hat{b}_t \right) + \lambda_2 \mathcal{L}_{GIoU} \left( b_t, \hat{b}_t \right) \right), \tag{20}$$

where $\lambda_1$ and $\lambda_2$ are regularization parameters, which are set as 5 and 2, respectively. $b_t$ denotes the $t$-th predicted bounding box, $\hat{b}_t$ is the corresponding ground truth. The overall loss can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{reg}. \tag{21}$$

## 4. Experiment

### *4.1. Datasets and Metrics*

To fully validate the effectiveness of our method, we evaluate it on existing representative multi-modal tracking tasks, including RGB-T tracking, RGB-D tracking, and RGB-E tracking. We use the standard evaluation metrics for each task to validate the performance of the proposed method.

**RGB-T tracking.** For RGB-T tracking, we evaluate our method on the latest RGB-T tracking datasets, RGBT234 [36] and LasHeR [6], which are the largest RGB-T tracking datasets with more than 200 testing video sequences of different challenging attributes. We use Precision Rate (PR) and Success Rate (SR) as primary measures, and the threshold of center location error is set to 20 pixels. PR represents the ratio of frames $f_p$ with center error smaller than a threshold to the total number of frames $N$, which can be calculated as:

$$PR = \frac{N_p}{N}, \tag{22}$$

where $N_p$ represents the number of frames with center error smaller than a threshold; $N$ represents the total number of frames in the sequence. SR is defined as the ratio of

frames $s_p$ with IoU exceeding a certain threshold to the total number of frames $N$, and it can be calculated as:

$$SR = \frac{N_s}{N}, \tag{23}$$

where $N_s$ represents the number of frames with IoU exceeding a certain threshold.

**RGB-D tracking.** For RGB-D tracking, we conduct experiments on the DepthTrack [11] and VOT22-RGBD [37] datasets. DepthTrack is a large-scale, long-term RGB-D tracking dataset consisting of 200 pairs of RGB-D videos, and the VOT-RGBD22 dataset is the latest RGB-D tracking dataset proposed in the VOT challenge [37], containing more than 140 test sequences. For evaluation on DepthTrack, we use Precision (Pre), Recall (Re), and F-score as metrics, while for VOT22-RGBD, we adopt Accuracy (A), Robustness (R), and Expected Average Overlap (EAO) to assess performance. Precision is calculated by the Gaussian Mixture Distribution between all frame output boxes and the given correct output boxes. The sum of all computed Gaussian Mixture Distributions is divided by the total frame count to determine tracking precision. The precision is calculated as follows:

$$\mathrm{Pre}\left(\tau_\theta\right) = \frac{1}{N_p} \sum_t \Omega\left(A_t\left(\theta_t\right), G_t\right), t \in \left\{t : A_t\left(\theta_t\right) \neq \emptyset\right\}, \tag{24}$$

where $\mathrm{Pre}\left(\tau_\theta\right)$ represents precision, $A_t\left(\theta_t\right)$ represents the tracker's output, $G_t$ represents the ground truth, and $\Omega(\cdot)$ represents the intersection of the two. The sum is taken over all non-empty predicted results.

Recall is calculated by the Gaussian Mixture Distribution between all frame output boxes and the given correct output boxes. The sum of all computed Gaussian Mixture Distributions is divided by the total frame count where targets are present to determine tracking recall:

$$\mathrm{Re}\left(\tau_\theta\right) = \frac{1}{N_g} \sum_t \Omega\left(A_t\left(\theta_t\right), G_t\right), t \in \left\{t : G_t \neq \emptyset\right\}, \tag{25}$$

where $\mathrm{Re}\left(\tau_\theta\right)$ represents recall, $A_t\left(\theta_t\right)$ represents the tracker's output. The sum is taken over all non-empty ground truth results.

F-score is divided by the summary of Pr and Re and then multiplied by two to obtain the tracking F-score:

$$F\text{-}score\left(\tau_\theta\right) = 2\frac{\mathrm{Pr}\left(\tau_\theta\right)\mathrm{Re}\left(\tau_\theta\right)}{\left(\mathrm{Pr}\left(\tau_\theta\right) + \mathrm{Re}\left(\tau_\theta\right)\right)}, \tag{26}$$

Table 1: Comparison between the proposed method and the state-of-the-art trackers on RGB-T datasets. The best results are highlighted in **bold**. The performance is evaluated in terms of Precision Rate (PR) and Success Rate (SR). The trackers that use temporal information are marked in *.

| | Methods | Publication | RGBT234 | | LasHeR | | Params(M) | FPS↑ |
|---|---|---|---|---|---|---|---|---|
| | | | PR↑ | SR↑ | PR↑ | SR↑ | | |
| Traditional | *TBSI [10] | CVPR23 | 0.871 | 0.637 | 0.692 | 0.556 | 350 | 36 |
| | *MTNet [9] | ICME23 | 0.850 | 0.619 | 0.608 | 0.474 | - | 55 |
| | GMMT [38] | AAAI24 | 0.879 | 0.647 | 0.707 | 0.566 | - | - |
| | *STMT [39] | TCSVT24 | 0.865 | 0.638 | 0.674 | 0.537 | - | 39 |
| | CAT++ [40] | TIP24 | 0.840 | 0.592 | 0.509 | 0.356 | 90 | 14 |
| | *ODTrack [30] | AAAI24 | 0.659 | 0.664 | 0.702 | 0.555 | - | 70 |
| | CAFormer [41] | AAAI25 | 0.867 | 0.648 | 0.700 | 0.556 | 93.4 | 84 |
| | UNIRTL [42] | PR25 | 0.773 | 0.594 | 0.587 | 0.454 | 99.1 | 58 |
| | SMGNet [43] | PR25 | 0.895 | 0.667 | 0.720 | **0.574** | 134.9 | 41 |
| Prompt | ProTrack [44] | MM23 | 0.795 | 0.599 | 0.538 | 0.420 | - | 30 |
| | ViPT [14] | CVPR23 | 0.835 | 0.617 | 0.651 | 0.525 | 93 | 25 |
| | *SDSTrack [13] | CVPR24 | 0.848 | 0.625 | 0.665 | 0.531 | 107.8 | 21 |
| | UN-Track [45] | CVPR24 | 0.837 | 0.618 | 0.667 | 0.536 | 92.1 | - |
| | OneTracker [15] | CVPR24 | 0.857 | 0.642 | 0.672 | 0.538 | 99.8 | - |
| | *TaTrack [46] | AAAI24 | 0.872 | 0.644 | 0.702 | 0.561 | - | 26 |
| | BaT [47] | AAAI24 | 0.868 | 0.641 | 0.702 | 0.563 | - | - |
| | IPL [48] | IJCV25 | 0.883 | 0.657 | 0.694 | 0.553 | - | - |
| | CMDTrack [49] | TPAMI25 | 0.859 | 0.618 | 0.688 | 0.566 | - | 67 |
| | **Ours** | - | **0.919** | **0.689** | **0.726** | 0.571 | 98.9 | 65 |

where Re ($\tau_\theta$) represents the corresponding recall; Pre ($\tau_\theta$) represents the corresponding precision.

**RGB-E tracking.** For RGB-E tracking, we report the experimental results on VisEvent [8], which is the largest RGB-E tracking dataset, containing over 500 sequences. The metrics that we use to evaluate are Precision Rate (PR) and Success Rate (SR), just the same as RGB-T tracking.

### *4.2. Experimental Settings*

When fine-tuning the proposed method, we choose the training sets of LasHeR for RGB-T tracking, DepthTrack for RGB-D tracking, and VisEvent for RGB-E tracking. The proposed method is trained on one NVIDIA RTX 4090 GPU with a batch size of 16. We use the ViT-base trained on MAE as our baseline. The training consists

Table 2: Comparison between the proposed method and the state-of-the-art trackers on RGB-D datasets. The best results are highlighted in **bold**. The performance is evaluated in terms of precision (Pre), recall (Re), F-score(F) on DepthTrack and EAO, accuracy(A), and robustness(R) on VOT-RGBD22. The trackers that use temporal information are marked in *.

| | Methods | Publication | DepthTrack | | | VOT-RGBD22 | | |
|---|---|---|---|---|---|---|---|---|
| | | | Pre↑ | Re↑ | F↑ | EAO↑ | A↑ | R↑ |
| Traditional | DeT [11] | ICCV21 | 0.506 | 0.560 | 0.532 | 0.657 | 0.760 | 0.845 |
| | SBT-D [37] | ECCV22 | - | - | - | 0.708 | 0.809 | 0.864 |
| | OSTrack [50] | ECCV22 | 0.536 | 0.522 | 0.529 | 0.676 | 0.803 | 0.833 |
| | SPT [7] | AAAI23 | 0.549 | 0.527 | 0.538 | 0.651 | 0.798 | 0.851 |
| | *ARKitTrack [25] | CVPR23 | 0.617 | 0.607 | 0.612 | - | - | - |
| | *ODTrack [30] | AAAI24 | 0.586 | 0.610 | 0.598 | - | - | - |
| | *TABBTrack [51] | PR25 | 0.622 | 0.615 | 0.618 | - | - | - |
| Prompt | ProTrack [44] | MM23 | 0.583 | 0.573 | 0.578 | 0.651 | 0.801 | 0.802 |
| | ViPT [14] | CVPR23 | 0.596 | 0.594 | 0.592 | 0.721 | 0.815 | 0.871 |
| | *SDSTrack [13] | CVPR24 | 0.619 | 0.609 | 0.614 | 0.728 | 0.812 | 0.883 |
| | UN-Track [45] | CVPR24 | 0.560 | 0.557 | 0.558 | 0.721 | 0.815 | 0.871 |
| | OneTracker [15] | CVPR24 | 0.609 | 0.604 | 0.607 | 0.721 | 0.819 | 0.872 |
| | CMDTrack [49] | TPAMI25 | 0.591 | 0.607 | 0.598 | - | - | - |
| | **Ours** | - | **0.636** | **0.663** | **0.649** | **0.773** | **0.821** | **0.933** |

of two stages: in the first stage, we fine-tune the visual adapter and patch-embedding layer for 60 epochs while freezing the other part of the network. In the second stage, we fine-tune the memory adapter on top of stage one for another 60 epochs and freeze the other part except for the patch-embedding layer and visual adapter. Each epoch contains 10,000 samples, and we use the AdamW optimizer with a learning rate of $5e^{-4}$. In addition, our method contains a total of 98.9M parameters, introducing 7.3M more than the baseline, with an additional computational cost of 1 GFlops.

### *4.3. Comparison with the State-of-the-Art Methods*

We compare the proposed method with two categories of state-of-the-art multi-modal trackers: traditional two-branch methods and prompt-learning methods. The former includes methods specifically designed for a particular type of multi-modal tracking that fully fine-tunes the two-branch network for both modalities. The latter

Table 3: Comparison between the proposed method and the state-of-the-art trackers on RGB-E datasets. The best results are highlighted in **bold**. The performance is evaluated in terms of precision rate (PR) and success rate (SR). The trackers that use temporal information are marked in *.

| | Methods | Publication | VisEvent | |
|---|---|---|---|---|
| | | | PR↑ | SR↑ |
| Traditional | Dimp [52] | ICCV19 | 0.691 | 0.533 |
| | TansT [46] | CVPR21 | 0.676 | 0.511 |
| | OSTrack [50] | ECCV22 | 0.695 | 0.534 |
| | SwinEFT [53] | AI23 | 0.710 | 0.565 |
| | *ODTrack [30] | AAAI24 | 0.727 | 0.553 |
| | MMHT [54] | NN24 | 0.732 | 0.551 |
| | CEUTrack [55] | PR25 | 0.691 | 0.531 |
| Prompt | ProTrack [44] | MM23 | 0.632 | 0.471 |
| | ViPT [14] | CVPR23 | 0.758 | 0.592 |
| | *SDSTrack [13] | CVPR24 | 0.767 | 0.597 |
| | UN-Track [45] | CVPR24 | 0.763 | 0.597 |
| | OneTracker [15] | CVPR24 | 0.767 | 0.608 |
| | CMDTrack [49] | TPAMI25 | 0.758 | 0.613 |
| | **Ours** | - | **0.803** | **0.626** |

encompasses methods designed for general multi-modal tracking that only fine-tune the adapters and patch-embedding layers.

**RGB-T tracking.** As reported in Table 1, the proposed method achieves superior performance against state-of-the-art trackers, including the recent SMGNet [43]. Specifically, on RGBT234, it surpasses the second-best SMGNet by 2.4% and 2.2% in PR and SR, respectively. On LasHeR, although SMGNet yields a marginally higher SR, our method relies on a much lighter architecture and runs significantly faster under the same hardware environment. Overall, our method provides the best trade-off between accuracy and efficiency, validating the effectiveness of the proposed prompt-learning framework. Furthermore, compared to methods employing online temporal updates, such as STMT [39] and TBSI [10], our method achieves superior accuracy without incurring high computational costs, validating that the memory adapter is more effective than traditional template updating in handling temporal variations.

**RGB-D tracking.** As shown in Table 2, our method achieves competitive perfor-

Table 4: Component analysis on multi-modal tracking datasets. Visual represents the proposed visual adapter, and memory represents the memory adapter.

| Visual | Memory | LasHeR | | DepthTrack | | | VisEvent | |
|---|---|---|---|---|---|---|---|---|
| | | PR↑ | SR↑ | Pre↑ | Re↑ | F↑ | PR↑ | SR↑ |
| | | 0.659 | 0.518 | 0.586 | 0.608 | 0.598 | 0.784 | 0.605 |
| ✓ | | 0.718 | 0.565 | 0.614 | 0.639 | 0.626 | 0.790 | 0.618 |
| | ✓ | 0.689 | 0.545 | 0.600 | 0.625 | 0.613 | 0.795 | 0.618 |
| ✓ | ✓ | **0.726** | **0.571** | **0.636** | **0.663** | **0.649** | **0.803** | **0.626** |

mance on DepthTrack and VOT22, which achieves 0.636, 0.663, and 0.649 in precision, recall, and F-score on the DepthTrack dataset, and 0.773, 0.821, and 0.933 in EAO, accuracy, and robustness on the VOT-RGBD22 dataset, respectively. This success is largely attributed to the proposed FMFA module, which effectively extracts low-frequency structural cues from depth images. These cues are critical for separating targets from cluttered backgrounds where color information alone is insufficient. Notably, our method significantly outperforms recent temporal-aware trackers. For instance, it surpasses the prompt-based online tracker SDSTrack [13] with gains of 5% in robustness and 4.5% in EAO on VOT22. This demonstrates that our multi-level memory retrieves more reliable temporal cues than standard frame-by-frame update strategies.

**RGB-Event tracking.** Table 3 reports the results on VisEvent. Our method sets a new state-of-the-art with 0.748 SR. Unlike traditional frames, event streams contain high-frequency motion information. Our frequency decomposition mechanism is naturally suited to capture these rapid changes, allowing the tracker to maintain robustness even during fast motion, where traditional RGB trackers often fail due to motion blur. Additionally, the proposed method achieves gains of 9.3% in PR and 6.1% in SR over SwinEFT [53], which is specifically designed for RGB-E tracking. These results demonstrate that our tracker also exhibits strong generalization capability in various tracking tasks.

Table 5: Effectiveness of visual adapter on the LasHeR dataset, the performance is evaluated in terms of precision rate (PR) and success rate (SR).

| Visual Adapter | Params(M) | PR↑ | SR↑ |
|---|---|---|---|
| - | 1.9 | 0.659 | 0.518 |
| Spatial | 2.1 | 0.690 | 0.544 |
| Spatial+Frequency | 3.3 | 0.701 | 0.551 |
| Spatial+Channel | 2.6 | 0.698 | 0.549 |
| Channel+Frequency | 3.6 | 0.700 | 0.552 |
| Spatial+Channel+Weighted Fusion | 2.7 | 0.707 | 0.565 |
| Spatial+Channel+Frequency | 3.8 | **0.718** | **0.565** |

### *4.4. Ablation Studies.*

*1) Effectiveness of Components:* We conduct an ablation study to evaluate key components of our method. As shown in Table 4, row 1 represents the baseline, which fuses RGB and auxiliary modalities via element-wise addition with fine-tuned patch embedding, yielding initial results. Replacing element-wise addition with our visual adapter significantly improves performance, notably on LasHeR, which improves 5.9% and 4.7% in PR and SR, respectively. On DepthTrack, this change lifts the F-score from 0.598 to 0.626, indicating that frequency-aware recalibration benefits sequences where depth maps are noisy but still contain high-frequency edge cues. This confirms that disentangling high and low-frequency components is crucial for mitigating cross-modal noise and bridging the modality gap. Integrating the multi-level memory into the baseline enhances robustness across all datasets. The gain is most evident on VisEvent, where SR rises from 0.605 to 0.618, confirming that temporal aggregation helps smooth the bursty event stream and recover from momentary information loss, particularly in robustifying the tracker against occlusion and long-term appearance changes. The full method, combining both components, achieves the highest performance gains, demonstrating their synergistic effectiveness in boosting tracking accuracy.

*2) Effectiveness of Visual Adapter:* To further validate the effectiveness of the proposed visual adapter, we conduct a comprehensive ablation study to investigate the contributions of its internal modules. The experimental results on the LasHeR dataset are presented in Table 5. The baseline performance without the Visual Adapter achieves

Table 6: Effectiveness of memory adapter on the LasHeR dataset, the performance is evaluated in terms of precision rate (PR) and success rate (SR).

| Memory Adapter | PR↑ | SR↑ |
|---|---|---|
| Propagate to adjacent | 0.659 | 0.518 |
| LSTM | 0.682 | 0.539 |
| Short Memory | 0.672 | 0.527 |
| Short+Long Memory | 0.680 | 0.537 |
| Short+Long+Permanent Memory | **0.689** | **0.545** |

a PR of 0.659 and an SR of 0.518. Incorporating spatial information alone significantly improves the metrics to 0.690 in PR and 0.544 in SR. Building upon the spatial features, independently adding either the frequency module or the channel module further enhances the representation, yielding PR scores of 0.701 and 0.698, respectively. To verify the importance of the fusion strategy, we introduced the weighted fusion mechanism, which significantly boosts the SR to 0.565 and demonstrates that adaptive feature selection is crucial for multi-modal alignment. Finally, integrating all three components, spatial, channel, and frequency modules, achieves the best overall performance with a PR of 0.718. This confirms that decomposing features into frequency subbands effectively isolates structural cues from noise, validating the necessity of the complete frequency-guided design.

*3) Effectiveness of Memory Adapter:* To further investigate the effectiveness of the Memory Adapter, the experimental results are shown in Table 6. The baseline method, employing ODTrack's adjacent-frame propagation, achieves PR/SR scores of 0.659/0.518. We explicitly compare our design with a recurrent update strategy using LSTM in row 2. While LSTM effectively models temporal dependencies and outperforms the baseline, it still falls short compared to our hierarchical design. Unlike parallel feature branches, our memory banks are designed as a hierarchically progressive that expands temporal coverage. Thus, we evaluate their contribution by gradually integrating them into the baseline. Our complete Memory Adapter, which integrates Short-term, Long-term, and Permanent memory banks, achieves the best performance with PR/SR scores of 0.689/0.545. These results demonstrate that explicitly storing and retrieving reliable historical tokens is more robust against long-term forgetting than

the latent state updates of LSTM, validating the superiority of our multi-level storage mechanism.

### *4.5. Parameter Analysis.*

Table 7: Memory configuration study on LasHeR. "S/L/P" denotes the number of tokens in **S**hort-, **L**ong- and **P**ermanent-term banks, respectively.

| Settings | Tokens (S/L/P) | PR↑ | SR↑ |
|---|---|---|---|
| – | 0/0/0 | 0.718 | 0.565 |
| Retrieval | 8/8/8 | 0.722 | 0.568 |
| Retrieval | 8/3/3 | 0.720 | 0.567 |
| Mean | 8/8/3 | 0.721 | 0.568 |
| **Retrieval** | **8/8/3** | **0.726** | **0.571** |

Table 8: Impact of template numbers and component effectiveness on LasHeR.

| Method | Templates | PR↑ | SR↑ |
|---|---|---|---|
| Baseline | 1 | 0.595 | 0.473 |
| Baseline | 3 | 0.659 | 0.518 |
| + Visual Adapter | 1 | 0.611 | 0.489 |
| + Visual Adapter | 3 | 0.718 | 0.565 |
| + Memory (Full) | 1 | 0.685 | 0.530 |
| **+ Memory (Full)** | **3** | **0.726** | **0.571** |

We investigate the impact of key hyper-parameters on the LasHeR dataset: the memory token configuration and the number of inference templates.

To determine the optimal memory structure, we evaluate different token sizes and retrieval strategies. As shown in Table 7, the baseline model without memory achieves PR of 0.718 and SR of 0.565. Enabling the memory module with a large bank of 8/8/8 tokens improves these metrics to 0.722 and 0.568, respectively. However, reducing the bank size to 3/3/3 diminishes the gains, implying that an undersized memory fails to capture sufficient appearance diversity. We also test replacing the similarity-based retrieval with naïve mean pooling. This yields results nearly identical to the large-bank setup, revealing that capacity alone is insufficient without an effective selection

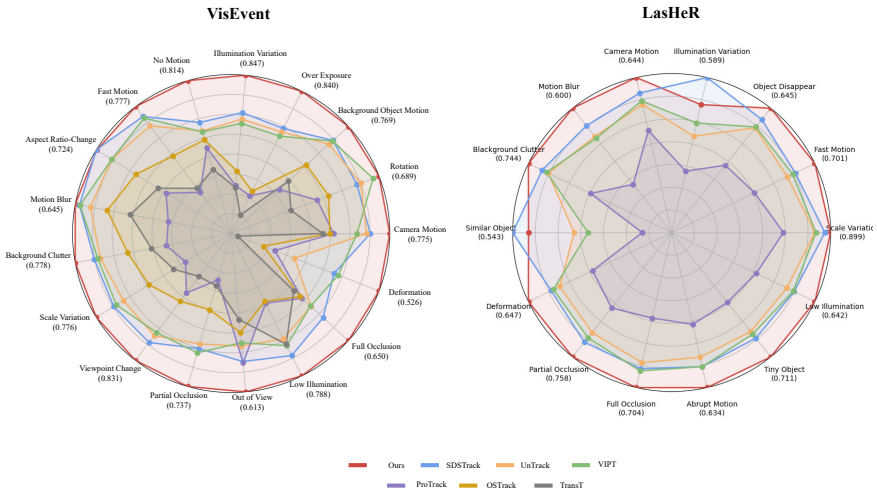


Figure 6: Precision scores of different attributes on the VisEvent and LasHeR.

mechanism. Finally, our default setting of 8/8/3 combined with the retrieval strategy achieves the best balance, ensuring that only the most pertinent memories influence the tracking.

We further analyze the robustness of our modules under different template settings. As shown in Table 8, while increasing template numbers generally improves performance, our proposed modules contribute significantly regardless of the template count. Specifically, under the challenging single-template setting, adding the Visual Adapter improves the SR from 0.473 to 0.489, and further integrating the Memory Adapter significantly boosts it to 0.530. This trend is consistent in the multi-template setting. This comparison demonstrates that the performance gains stem from the inherent effectiveness of our Frequency and Memory Adapters in feature learning and context modeling, rather than merely relying on multiple templates.

### *4.6. Attribute Analysis.*

We conduct a comprehensive attribute-based analysis on the LasHeR and VisEvent datasets. The evaluated attributes include Illumination Variation (IV), Thermal Crossover (TC), Partial/Full Occlusion (PO/FO), Motion Blur (MB), Fast Motion (FM), Background Clutter (BC), Low Illumination (LI), and Out-of-View (OV), as illustrated

Table 9: Comparison of tunable parameters on the LasHeR dataset, the performance is evaluated in terms of precision rate (PR) and success rate (SR).

| Method | Model Params(M) | Tunable Params(M) | LasHeR PR↑ | LasHeR SR↑ |
|---|---|---|---|---|
| Frozen | 93.4 | 1.9 | 0.659 | 0.518 |
| FFT | 93.4 | 93.4 | 0.702 | 0.555 |
| w/o Memory | 95.3 | 3.8 | 0.718 | 0.565 |
| **Ours** | 98.9 | 7.3 | **0.726** | **0.572** |

in Figure 6.

The radar charts demonstrate that our tracker encloses a consistently larger area than competing methods across almost all attributes. This indicates that our method achieves broad and robust improvements rather than isolated wins on specific scenarios. First, on attributes related to environmental interference, such as thermal crossover, our method exhibits a significant performance margin. In these scenarios, single-modality trackers often fail due to distinct degradations. Our gains are attributed to the frequency-guided visual adapter, which dynamically up-weights the reliable frequency components while suppressing modality-specific noise. Second, regarding target-specific challenges like full occlusion and out-of-view, our method outperforms existing trackers by a large margin. While conventional local trackers drift when the target disappears, our multi-level memory adapter effectively maintains a global context. The long-term and permanent memory banks enable the tracker to re-localize the target accurately upon re-appearance, ensuring stable tracking even after long-duration interruptions.

### *4.7. Comparison of Tunable Parameter.*

To evaluate the impact of different components on the tunable parameter and model complexity, we conduct a comparison of tunable parameters on the LasHeR dataset. As shown in Table 9, the first row represents the baseline model, where all components except the patch-embedding layer are frozen, resulting in a total of 93.4M parameters, with 1.9M being tunable. The second row shows the result of making all components of the baseline tunable, which improves the tracking performance by 4.3% in precision

Table 10: Comparison of visual adapters on multi-modal tracking datasets.

| Visual | Params(M) | LasHeR | | DepthTrack | | | VisEvent | |
|---|---|---|---|---|---|---|---|---|
| | | PR↑ | SR↑ | Pre↑ | Re↑ | F↑ | PR↑ | SR↑ |
| - | 1.9 | 0.659 | 0.518 | 0.586 | 0.608 | 0.598 | 0.784 | 0.605 |
| Fovea | 2.1 | 0.690 | 0.544 | 0.593 | 0.614 | 0.604 | 0.790 | 0.610 |
| **Ours** | 3.8 | **0.718** | **0.565** | **0.600** | **0.625** | **0.613** | **0.795** | **0.618** |

rate and 3.7% in SR; however, the number of tunable parameters rises to 93.4M. In the third row, the proposed visual adapter is added to the baseline and made learnable, adding only 1.9M tunable parameters, while improving the tracking performance by 5.9% in precision rate and 4.7% in success rate, compared to the baseline. The last row shows the final version of the proposed method, which has 98.9M model parameters, and 7.3M of them are tunable. In addition, our method introduces about 1 GFlops computational cost more than the baseline.

### *4.8. Comparison of Visual Adapters.*

We conduct comparison experiments between our proposed visual adapter and the widely adopted Fovea fusion adapter [14, 31, 46], fine-tuning both under identical settings. The "–" row in Table 10 represents a strong baseline without any dedicated adapter. Introducing the Fovea fusion adapter improves tracking performance on the LasHeR dataset, with PR and SR increasing from 0.659 and 0.518 to 0.690 and 0.544, respectively. Similar trends are observed on DepthTrack and VisEvent, where Fovea yields modest but consistent gains, indicating that spatial re-weighting of features contributes positively to performance. Replacing Fovea with our frequency-aware visual adapter leads to further improvements, achieving 0.718 precision and 0.565 success on LasHeR. This represents absolute gains of +5.9% and +4.7% over the baseline on PR and SR, respectively, and +2.8% / +2.1% over Fovea. A similar trend is seen on DepthTrack, where the F-score improves from 0.598 to 0.613. On VisEvent, precision rises from 0.784 to 0.795. Although our module introduces 3.8 M tunable parameters in total, this is only 1.7 M more than the Fovea variant and remains lightweight compared with full fine-tuning of the backbone. The extra capacity is used to model frequency,

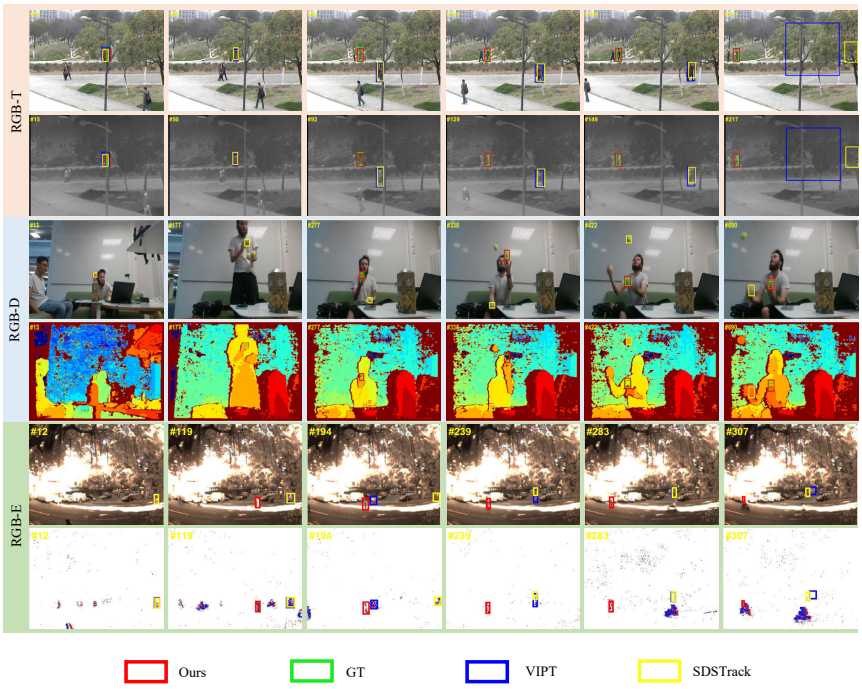


Figure 7: Qualitative comparison of our method with ViPT and SDSTrack on RGB-T, RGB-D, and RGB-E tracking benchmarks.

domain interactions across RGB and auxiliary modalities, which explains why the gain is most pronounced on LasHeR, where motion blur and illumination variation make high-frequency cues especially informative, while still yielding steady improvements on the other two datasets. In sum, Table 10 confirms that our adapter offers the best accuracy–efficiency trade-off among the tested designs, outperforming both the plain baseline and the strong Fovea fusion across all metrics without incurring a prohibitive parameter cost.

### *4.9. Qualitative Comparison and Feature Analysis.*

*1) Qualitative Comparison with SOTA Methods:* To intuitively demonstrate the effectiveness of our approach, we visualize tracking responses alongside two strong SOTA baselines, ViPT and SDSTrack, in Figure 7.

The top row shows a representative RGB-T sequence with severe occlusions and

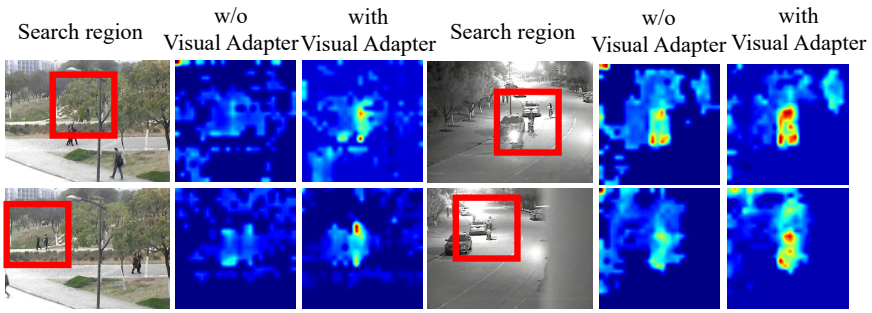


Figure 8: Feature visualization before and after applying the visual adapter.

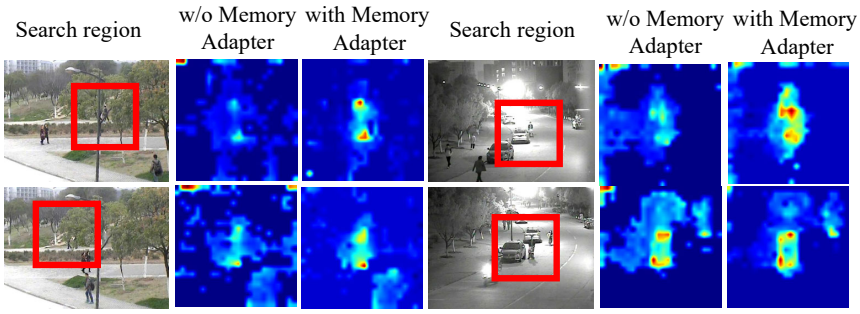


Figure 9: Feature visualization before and after applying the memory adapter.

multiple visually similar distractors. While both baselines quickly drift off the target, our tracker consistently maintains focus, benefiting from the memory adapter's ability to preserve a reliable temporal trajectory. The middle row presents a fast-motion RGB-D sequence containing several similar instances. Despite rapid camera and object movement, our method robustly localizes the correct target, highlighting its adaptability to depth-aided motion scenarios. The bottom row illustrates a challenging RGB-E case with cluttered, low-resolution RGB frames and clean, high-temporal-resolution event data. By effectively fusing these complementary modalities, the visual adapter generates sharp and unambiguous response maps, enabling accurate tracking even under complex background interference.

*2) Feature Representation Analysis:* To provide a more comprehensive visual analysis, we further inspect the feature visualization produced by both the visual and memory adapters. For the visual adapter, Figure 8 arranges three columns: the left col-

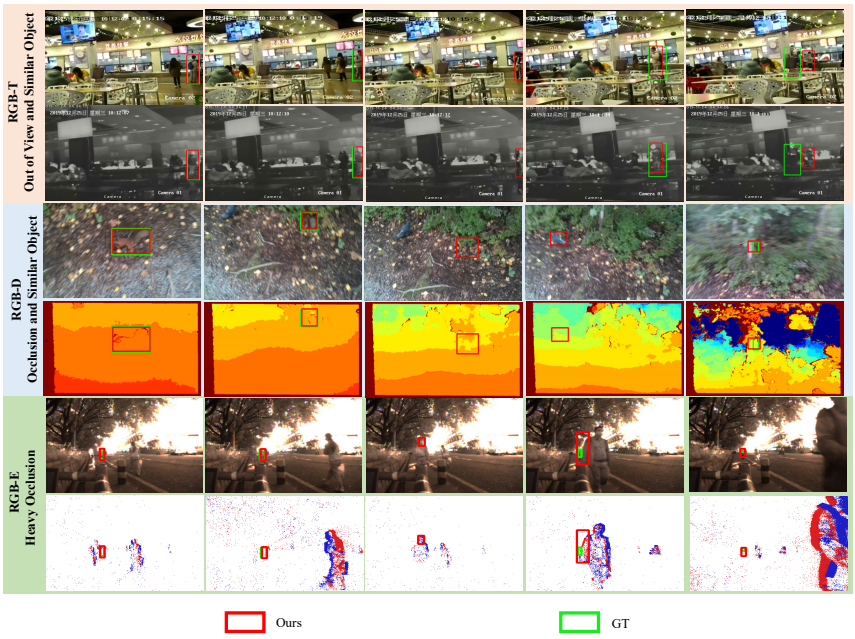


Figure 10: Visualization of representative failure cases.

umn shows the raw search region, the middle column depicts the feature map generated by the baseline without our adapter, and the right column presents the features after using the adapter. Once the adapter is enabled, the target contour becomes crisp, and the clutter in the background is markedly suppressed, confirming that frequency-aware recalibration strengthens discriminative cues while filtering out irrelevant noise. The memory adapter cannot be plotted directly because it stores one-dimensional vectors, so we instead compare the target-region feature maps before and after the memory adapter in Figure 9. After memory aggregation, the foreground activations are noticeably sharper, and the separation from the background is clearer, illustrating that temporal information preserved in memory further refines spatial representations and stabilizes tracking.

### *4.10. Limitations and Failure Analysis*

To further investigate the performance bottlenecks of our proposed method, we present a visualization of representative failure cases across RGB-T, RGB-D, and RGB-

E benchmarks in Figure 10. The primary failure mode occurs under heavy occlusion or out-of-view scenarios. For instance, in the RGB-D case, when the target is completely occluded by a pedestrian, the tracker loses valid visual cues from both modalities and drifts towards a nearby similar distractor. Similarly, in the RGB-E and RGB-T cases, heavy occlusion or moving out of the view causes temporary tracking loss. Although our multi-level memory adapter enables the tracker to re-localize the target upon re-appearance in many cases, maintaining stable tracking during long-time occlusion remains a significant challenge. In these extreme situations, visual features become unreliable, and the tracker may struggle to distinguish the target from background clutter without explicit motion modeling.

## 5. Conclusion

In this paper, we proposed a novel prompt-learning framework that learns frequency- and memory-aware prompts for multi-modal object tracking. The frequency-guided visual adapter adaptively fuses auxiliary modalities with RGB by exploiting complementary spatial, channel, and frequency information, while the multi-level memory adapter stores and retrieves temporal cues to ensure consistent long-range propagation across frames. This dual-adapter design enables efficient and robust prompt learning on frozen trackers. Extensive experiments on RGB-T, RGB-D, and RGB-E benchmarks demonstrate that our method achieves state-of-the-art performance, significantly outperforming both fully fine-tuned and adapter-based baselines.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (62072232, 62576098), the Key R&D Project of Jiangsu Province (BE2022138), the Fundamental Research Funds for the Central Universities (021714380026), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

## References


[1] X. Li, B. Zhong, Q. Liang, Z. Mo, J. Nong, S. Song, Dynamic updates for language adaptation in visual–language tracking, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2025, pp. 19165–19174.

[2] M. Feng, J. Su, Rgb-t tracking: A comprehensive review, Inf. Fusion (2024) 102492.

[3] P. Zhang, J. Zhao, D. Wang, H. Lu, X. Ruan, Visible-thermal uav tracking: A large-scale benchmark and new baseline, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022.

[4] Y. Li, L. Zheng, Y. Wang, E. Dong, S. Zhang, Impedance learning-based adaptive force tracking for robot on unknown terrains, IEEE Trans. Robot. (2025).

[5] Y. Zheng, B. Zhong, Q. Liang, G. Li, R. Ji, X. Li, Towards unified token learning for vision–language tracking, IEEE Trans. Circuits Syst. Video Technol. (2023).

[6] C. Li, W. Xue, Y. Jia, Z. Qu, B. Luo, J. Tang, D. Sun, Lasher: A large-scale high-diversity benchmark for rgb-t tracking, IEEE Trans. Image Process. 31 (2021) 392–404.

[7] X.-F. Zhu, T. Xu, Z. Tang, Z. Wu, H. Liu, X. Yang, X.-J. Wu, J. Kittler, Rgbd1k: A large-scale dataset and benchmark for rgb-d object tracking, in: Proc. AAAI Conf. Artif. Intell., 2023.

[8] X. Wang, J. Li, L. Zhu, Z. Zhang, Z. Chen, X. Li, Y. Wang, Y. Tian, F. Wu, Visevent: Reliable object tracking via collaboration of frame and event flows, IEEE Trans. Cybern. (2023).

[9] R. Hou, B. Xu, T. Ren, G. Wu, Mtnet: Learning modality-aware representation with transformer for rgb-t tracking, in: Proc. IEEE Int. Conf. Multimedia Expo, 2023.

[10] T. Hui, Z. Xun, F. Peng, J. Huang, X. Wei, X. Wei, J. Dai, J. Han, S. Liu, Bridging search region interaction with template for rgb-t tracking, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2023.

[11] S. Yan, J. Yang, J. Käpylä, F. Zheng, A. Leonardis, J. K. Kärkkäinen, Depthtrack: Unveiling the power of rgb-d tracking, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021.

[12] R. Hou, T. Ren, G. Wu, Mirnet: A robust rgbt tracking jointly with multi-modal interaction and refinement, in: Proc. IEEE Int. Conf. Multimedia Expo, 2022, pp. 1–6.

[13] X. Hou, J. Xing, Y. Qian, Y. Guo, S. Xin, J. Chen, K. Tang, M. Wang, Z. Jiang, L. Liu, et al., Sdstrack: Self-distillation symmetric adapter learning for multi-modal visual object tracking, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2024.

[14] J. Zhu, S. Lai, X. Chen, D. Wang, H. Lu, Visual prompt multi-modal tracking, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2023.

[15] L. Hong, S. Yan, R. Zhang, W. Li, X. Zhou, P. Guo, K. Jiang, Y. Chen, J. Li, Z. Chen, et al., Onetracker: Unifying visual object tracking with foundation models and efficient tuning, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2024.

[16] Y. Cui, W. Ren, X. Cao, A. Knoll, Image restoration via frequency selection, IEEE Trans. Pattern Anal. Mach. Intell. (2023).

[17] H. Zhou, C. Tian, Z. Zhang, C. Li, Y. Xie, Z. Li, Frequency-aware feature aggregation network with dual-task consistency for rgb-t salient object detection, Pattern Recogn. 146 (2024) 110043.

[18] S. Fan, X. Chen, C. He, L. Yu, Z. Mao, Y. Zheng, Multiple frequency–spatial network for rgb-t tracking in the presence of motion blur, Neural Comput. Appl. 35 (34) (2023) 24389–24406.

[19] J. Mei, J. Zhou, J. Wang, J. Hao, D. Zhou, J. Cao, Learning multi-frequency integration network for rgb-t tracking, IEEE Sensors J. (2024).

[20] Z. Cao, Z. Huang, L. Pan, S. Zhang, Z. Liu, C. Fu, Tctrack: Temporal contexts for aerial tracking, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022.

[21] T. Yang, A. B. Chan, Visual tracking via dynamic memory networks, IEEE Trans. Pattern Anal. Mach. Intell. 43 (1) (2021) 360–374.

[22] D. Yuan, X. Shu, Q. Liu, Z. He, Aligned spatial–temporal memory network for thermal infrared target tracking, IEEE Trans. Circuits Syst. II Express Briefs 70 (3) (2023) 1224–1228.

[23] H. K. Cheng, A. G. Schwing, Xmem: Long-term video object segmentation with an atkinson–shiffrin memory model, in: Proc. Eur. Conf. Comput. Vis., 2022.

[24] H. Ebbinghaus, Memory: A contribution to experimental psychology, Ann. Neurosci. 20 (4) (2013) 155.

[25] H. Zhao, J. Chen, L. Wang, H. Lu, Arkittrack: A new diverse dataset for tracking using mobile rgb-d data, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2023.

[26] Y. Xiao, M. Yang, C. Li, L. Liu, J. Tang, Attribute-based progressive fusion network for rgb-t tracking, in: Proc. AAAI Conf. Artif. Intell., 2022.

[27] X. Chen, B. Kang, W. Geng, J. Zhu, Y. Liu, D. Wang, H. Lu, Sutrack: Towards simple and unified single object tracking, in: Proc. AAAI Conf. Artif. Intell., Vol. 39, 2025, pp. 2239–2247.

[28] B. Yan, H. Peng, J. Fu, D. Wang, H. Lu, Learning spatio–temporal transformer for visual tracking, in: Proc. IEEE Int. Conf. Comput. Vis., 2021.

[29] X. Chen, H. Peng, D. Wang, H. Lu, H. Hu, Seqtrack: Sequence-to-sequence learning for visual object tracking, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2023.

[30] Y. Zheng, B. Zhong, Q. Liang, Z. Mo, S. Zhang, X. Li, Odtrack: Online dense temporal token learning for visual tracking, in: Proc. AAAI Conf. Artif. Intell., 2024.

[31] X. Chen, B. Kang, J. Zhu, D. Wang, H. Peng, H. Lu, Unified sequence-to-sequence learning for single- and multi-modal visual object tracking, arXiv:2404.00000 (2024).

[32] B. Yan, H. Peng, K. Wu, D. Wang, J. Fu, H. Lu, Lighttrack: Finding lightweight neural networks for object tracking via one-shot architecture search, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 15180–15189.

[33] P. Blatter, M. Kanakis, M. Danelljan, L. Van Gool, Efficient visual tracking with exemplar transformers, in: Proceedings of the IEEE/CVF Winter conference on applications of computer vision, 2023, pp. 1571–1581.

[34] B. Kang, X. Chen, D. Wang, H. Peng, H. Lu, Exploring lightweight hierarchical vision transformers for efficient visual tracking, in: Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 9612–9621.

[35] V. Borsuk, R. Vei, O. Kupyn, T. Martyniuk, I. Krashenyi, J. Matas, Fear: Fast, efficient, accurate and robust visual tracker, in: European conference on computer vision, Springer, 2022, pp. 644–663.

[36] C. Li, X. Liang, Y. Lu, N. Zhao, J. Tang, Rgb-t object tracking: Benchmark and baseline, Pattern Recogn. 96 (2019) 106977.

[37] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, et al., The tenth visual object tracking vot2022 challenge results, in: Proc. Eur. Conf. Comput. Vis., 2022.

[38] Z. Tang, T. Xu, X. Wu, X.-F. Zhu, J. Kittler, Generative-based fusion mechanism for multi-modal tracking, in: Proc. AAAI Conf. Artif. Intell., 2024.

[39] D. Sun, Y. Pan, A. Lu, C. Li, B. Luo, Transformer rgb-t tracking with spatio-temporal multimodal tokens, IEEE Trans. Circuits Syst. Video Technol. 34 (11) (2024) 12059–12072.

[40] L. Liu, C. Li, Y. Xiao, R. Ruan, M. Fan, Rgb-t tracking via challenge-based appearance disentanglement and interaction, IEEE Trans. Image Process. (2024).

[41] Y. Xiao, J. Zhao, A. Lu, C. Li, B. Yin, Y. Lin, C. Liu, Cross-modulated attention transformer for rgb-t tracking, in: Proc. AAAI Conf. Artif. Intell., Vol. 39, 2025, pp. 8682–8690.

[42] L. Zhang, L. Wang, Y. Wu, M. Chen, D. Zheng, L. Cao, B. Zeng, Y. Cai, Unirtl: A universal rgbt and low-light benchmark for object tracking, Pattern Recognition 158 (2025) 110984.

[43] L. Liu, C. Li, J. Tang, C. Li, Rgbt tracking via supervised mutual guiding, Pattern Recognition (2025) 112295.

[44] J. Yang, Z. Li, F. Zheng, A. Leonardis, J. Song, Prompting for multi-modal tracking, in: Proc. ACM Int. Conf. Multimedia, 2022.

[45] Z. Wu, J. Zheng, X. Ren, F.-A. Vasluianu, C. Ma, D. P. Paudel, L. Van Gool, R. Timofte, Single-model and any-modality for video object tracking, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2024.

[46] H. Wang, X. Liu, Y. Li, M. Sun, D. Yuan, J. Liu, Temporal adaptive rgb-t tracking with modality prompt, in: Proc. AAAI Conf. Artif. Intell., 2024.

[47] B. Cao, J. Guo, P. Zhu, Q. Hu, Bi-directional adapter for multimodal tracking, in: Proc. AAAI Conf. Artif. Intell., 2024.

[48] A. Lu, C. Li, J. Zhao, J. Tang, B. Luo, Modality-missing rgb-t tracking: Invertible prompt learning and high-quality benchmarks, Int. J. Comput. Vis. 133 (5) (2025) 2599–2619.

[49] T. Zhang, Q. Zhang, K. Debattista, J. Han, Cross-modality distillation for multi-modal tracking, IEEE Trans. Pattern Anal. Mach. Intell. 47 (7) (2025) 5847–5865.

[50] B. Ye, H. Chang, B. Ma, S. Shan, X. Chen, Joint feature learning and relation modeling for tracking: A one-stream framework, in: Proc. Eur. Conf. Comput. Vis., 2022.

[51] G. Ying, D. Zhang, Z. Ou, X. Wang, Z. Zheng, Temporal adaptive bidirectional bridging for rgb-d tracking, Pattern Recogn. 158 (2025) 111053.

[52] G. Bhat, M. Danelljan, L. Van Gool, R. Timofte, Learning discriminative model prediction for tracking, in: Proc. IEEE Int. Conf. Comput. Vis., 2019.

[53] Z. Zeng, X. Li, C. Fan, L. Zou, R. Chi, Swineft: A robust and powerful swin transformer based event frame tracker, Appl. Intell. 53 (20) (2023) 23564–23581.

[54] H. Sun, R. Liu, W. Cai, J. Wang, Y. Wang, H. Tang, Y. Cui, D. Yao, D. Guo, Reliable object tracking by multimodal hybrid feature extraction and transformer-based fusion, Neural Networks 178 (2024) 106493.

[55] C. Tang, X. Wang, J. Huang, B. Jiang, L. Zhu, S. Chen, J. Zhang, Y. Wang, Y. Tian, Revisiting color-event based tracking: A unified network, dataset, and metric, Pattern Recognition (2025) 112718.