SUTrack: Towards Simple and Unified Single Object Tracking

Xin Chen[1], **Ben Kang**[1], **Wanting Geng**[1], **Jiawen Zhu**[1], **Yi Liu**[2], **Dong Wang**[1*], **Huchuan Lu**[1]

[1] Dalian University of Technology
[2] Baidu Inc

chenxin3131@mail.dlut.edu.cn, kangben@mail.dlut.edu.cn, gengwanting@mail.dlut.edu.cn, jiawen@mail.dlut.edu.cn, liuyi22@baidu.com, wdice@dlut.edu.cn, lhchan@dlut.edu.cn

Abstract

In this paper, we propose a simple yet unified single object tracking (SOT) framework, dubbed SUTrack. It consolidates five SOT tasks (RGB-based, RGB-Depth, RGB-Thermal, RGB-Event, RGB-Language Tracking) into a unified model trained in a single session. Due to the distinct nature of the data, current methods typically design individual architectures and train separate models for each task. This fragmentation results in redundant training processes, repetitive technological innovations, and limited cross-modal knowledge sharing. In contrast, SUTrack demonstrates that a single model with a unified input representation can effectively handle various SOT tasks, eliminating the need for task-specific designs and separate training sessions. Additionally, we introduce a task-recognition training strategy and a soft token type embedding to further enhance SUTrack's performance with minimal overhead. Experiments show that SUTrack outperforms previous task-specific counterparts across 11 datasets spanning five SOT tasks. Moreover, we provide a range of models catering edge devices as well as high-performance GPUs, striking a good trade-off between speed and accuracy. We hope SUTrack could serve as a strong foundation for further compelling research into unified tracking models.

Code — github.com/chenxin-dlut/SUTrack

Introduction

Single object tracking (SOT) is a fundamental task in computer vision, focusing on locating an arbitrary target within a video sequence, starting from its initial location. Over the years, to broaden the application scenarios of SOT (Li et al. 2018; Bhat et al. 2019; Zhang et al. 2020; Chen et al. 2021; Wang et al. 2021a; Yan et al. 2021a; Ye et al. 2022; Wei et al. 2023; Zheng et al. 2024), numerous downstream SOT tasks incorporating auxiliary input modalities have been proposed. These tasks include RGB-Depth (Zhu et al. 2023b; Yan et al. 2021c), RGB-Thermal (Li et al. 2021, 2019b), RGB-Event (Wang et al. 2024), and RGB-Language (Wang et al. 2021b; Li et al. 2017) tracking. Existing SOT methods are characterized by fragmentation, with most approaches focusing on one or a few specific downstream tasks and developing separate models for each.

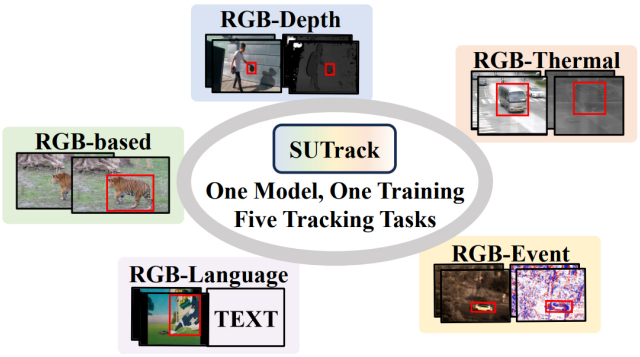

Figure 1: Our SUTrack unifies five SOT tasks into one model with one training session.

This fragmentation enables customized designs for each task, making it a prevalent choice. However, several deficiencies persist: First, each task requires training a separate model, resulting in redundant parameters and inefficient use of resources. Second, models are trained on task-specific datasets, which hinders the sharing of knowledge across all available datasets and increases the risk of overfitting. Third, technological innovations are often repeatedly designed and validated across different tasks, leading to duplicated efforts. Although some approaches to unify SOT tasks have emerged, their level of unification remains limited. For instance, some approaches (Zhu et al. 2023a; Hong et al. 2024; Hou et al. 2024) unify only the architectural design, not the model parameters, while others (Wu et al. 2024) address only a subset of tasks. This naturally raises the question: *Can a unified visual model address mainstream SOT tasks?*

To explore this question, we propose a simple and unified framework for SOT, named SUTrack. SUTrack unifies five mainstream SOT tasks: RGB-based, RGB-Depth, RGB-Thermal, RGB-Event, and RGB-Language tracking. It is based on a straightforward one-stream tracking architecture (Ye et al. 2022; Cui et al. 2022). By making concise improvements to the interface to accommodate various modalities, SUTrack achieves unification with a single model and a single training session. The underlying intuition is that modern general visual models should inherently be capable of integrating knowledge from different modalities. We simply

*Corresponding author.

need to convert these modalities into a unified form to train the model, rather than developing separate models for each modality.

To this end, we convert the RGB, depth, thermal, event, and language modalities into a unified token format for input into the vision transformer. Specifically, the depth, thermal, and event modalities are typically paired with the RGB modality in image format. Therefore, we modify the patch embedding layer of the vision transformer from three channels to six channels to accommodate channel-concatenated RGB-Depth, RGB-Thermal, or RGB-Event image pairs. These image pairs are converted into token embeddings by the modified patch embedding layer and can then be directly fed into the transformer. Unlike prevalent methods that employ additional branches to receive auxiliary modalities, this approach is more efficient, adding only 0.06 M parameters and less than 0.7 GFlops compared to a purely RGB-based tracker. For the language modality, we employ a CLIP (Radford et al. 2021) text encoder to convert the language input into a token embedding. We adopt a vision transformer to process these tokens, followed by a common center-based tracking head (Ye et al. 2022) to predict the result.

Additionally, we introduce a task-recognition auxiliary training strategy. Alongside standard tracking supervision, this approach involves classifying the source task of the input data during training. We found that incorporating this task-specific information enhances performance. Importantly, this strategy is used only during training and does not add any overhead during inference. Furthermore, the cropped template and search region can potentially cause confusion regarding token types (template background, template foreground, and search region) (Lin et al. 2024), especially for depth, thermal, and event data, which are typically less detailed than RGB data. To address this issue, We develop a soft token type embedding, drawing inspiration from the token type embedding introduced in LoRAT (Lin et al. 2024). This enhancement equips the model with more precise token type information.

Experiments demonstrate that our SUTrack method is effective, achieving new state-of-the-art performance across 11 benchmarks and five SOT tasks. For instance, SUTrack-B384 attains 74.4% AUC on the RGB-based benchmark LaSOT, surpassing the recent ODTrack-B384 (Zheng et al. 2024) by 1.2% while maintaining a similar model size. Moreover, when compared to recent multi-modal trackers (Hong et al. 2024; Hou et al. 2024), SUTrack consistently outperforms them across all evaluated datasets. It is worth noting that all these prior methods either train different models for each task or cannot cover all five SOT tasks, whereas our SUTrack handles all tasks with a unified model.

In summary, the contributions of this work are two-fold:

- We propose a simple yet unified SOT framework. It consolidates five SOT tasks into a unified model and learning paradigm. We believe this achievement will significantly reduce the research complexity across SOT tasks.
- We present a new family of unified tracking models that strike a good balance between speed and accuracy. Experiments confirm the effectiveness of these new models.

Related Work

RGB-based Object Tracking

RGB-based object tracking refers to SOT using only RGB data, typically serving as the foundation for downstream SOT tasks. RGB-based object tracking has witnessed significant progress (Tao, Gavves, and Smeulders 2016; Bertinetto et al. 2016; Nam and Han 2016; Danelljan et al. 2017, 2019; Li et al. 2019a; Xu et al. 2020; Mayer et al. 2021; Chen et al. 2021; Xie et al. 2022; Mayer et al. 2022; Lin et al. 2022) over the years, driven by advancements in deep models (Dosovitskiy et al. 2020; Radford et al. 2021).

Recently, one-stream transformer-based trackers (Cui et al. 2022; Ye et al. 2022; Chen et al. 2022a; Wu et al. 2023; Cai et al. 2023; Xie et al. 2023; He et al. 2023; Song et al. 2023) have initiated a new revolution in RGB-based object tracking. This framework more thoroughly utilizes the capabilities of pretrained transformers by jointly performing feature extraction and fusion, achieving new leading performance. Building on these pioneering works, we advance further in this paper by developing a new one-stream unified tracking framework through simple modifications to the input interface and training strategy. Our framework not only handles RGB-based object tracking tasks effectively but also performs multi-modal downstream SOT tasks simultaneously, showcasing the greater potential of the one-stream framework combined with modern pretrained transformer models.

Multi-Modal Object Tracking

To address the challenges faced by RGB-based tracking in complex or specific scenarios, multi-modal tracking tasks and methods (Zhu et al. 2023b; Feng et al. 2021; Yang et al. 2022) have been proposed. These tasks integrate auxiliary modalities beyond the RGB input, expanding the applicability of tracking algorithms. Common multi-modal tracking tasks now include RGB-Depth, RGB-Thermal, RGB-Event, and RGB-Language tracking. By incorporating depth (Yan et al. 2021c; Qian et al. 2021), thermal (Li et al. 2021; Xiao et al. 2022), event (Wang et al. 2024), or language (Wang et al. 2021b; Feng et al. 2021; Ma and Wu 2021) information, multi-modal trackers significantly enhance their ability to tackle issues such as occlusions, low lighting, extreme weather, and target variations.

Despite their impressive performance, existing multi-modal methods typically rely on modality-specific designs and training, *i.e.*, developing different models for each modality. This situation leads to inefficient use of data, computational resources, and human effort. In contrast, our approach integrates all multi-modal tracking tasks into a single, unified model and training paradigm. With just one training session, this unified model efficiently handles multiple multi-modal tracking tasks and achieves new state-of-the-art performance across these tasks.

Unified Object Tracking Models

With the advancement of foundational models (Dosovitskiy et al. 2020; Radford et al. 2021), it has become feasible to use unified frameworks or models to address multiple

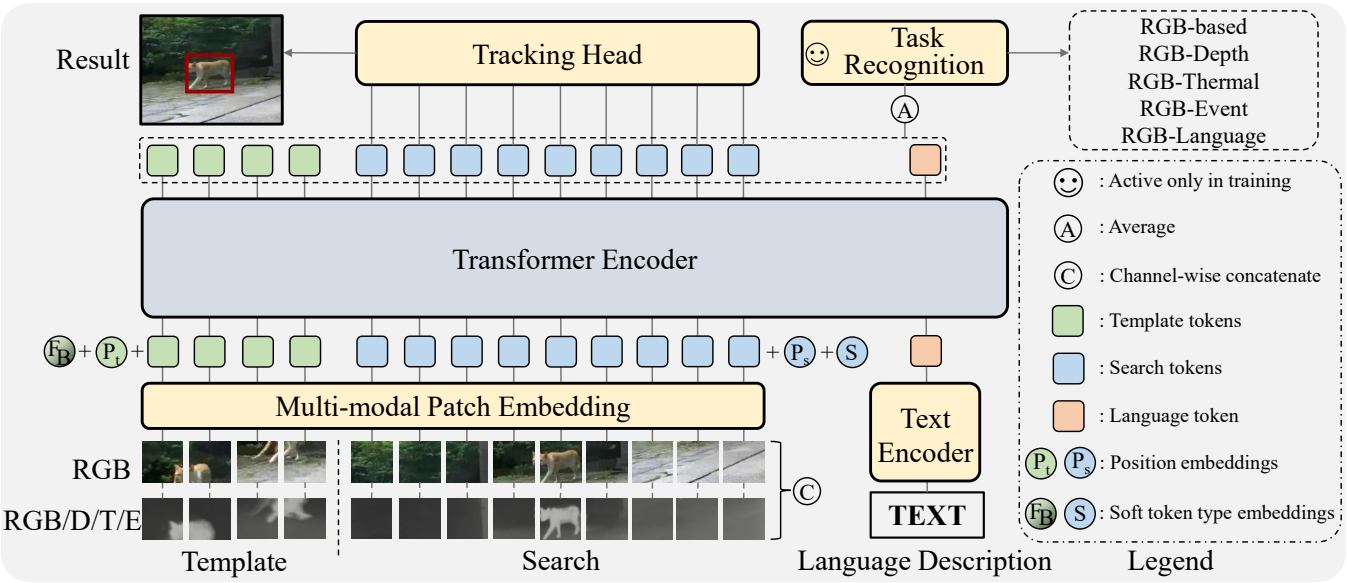

Figure 2: Architecture of the proposed SUTrack. SUTrack unifies five SOT tasks (RGB-based, RGB-Depth, RGB-Thermal, RGB-Event, RGB-Language Tracking) into a single model. We use a unified token embedding format to represent different modalities and train a transformer-based tracking model with these embeddings. In the figure, D/T/E denote depth, thermal, and event modalities, respectively.

tasks. Recently, several works have emerged that aim to unify multiple SOT tasks. ViPT (Zhu et al. 2023a) addresses three multi-modal tasks (RGB-Depth, RGB-Thermal, and RGB-Event) within a unified framework using prompt learning, but it does not achieve model-level unification. Subsequently, SDSTrack (Hou et al. 2024) achieves framework-level unification, while un-track (Wu et al. 2024) realizes model-level unification for these three tasks. OneTracker (Hong et al. 2024) unifies more tasks within a unified framework but does not achieve model-level unification. Despite these significant strides towards SOT unification, their level of unification remains incomplete. In this paper, our SUTrack, for the first time, integrates all five common SOT tasks into a single, streamlined model, further advancing the unification of SOT tasks.

SUTrack

The overall framework of SUTrack is illustrated in Fig. 2. It adopts a streamlined one-stream transformer architecture. First, input data from various modalities (including RGB, depth, thermal, event, and natural language) are converted into a unified embedding form. This unified representation enables the model to be trained to handle multiple SOT tasks. Next, positional embeddings and the proposed soft token type embeddings are added to the unified embeddings, enhancing positional information and providing precise prior knowledge about the token type (background/foreground). The vision transformer encoder then processes and associates these embeddings jointly. The resulting feature embeddings are used to support the final predictions, which are implemented using a center-based tracking head (Ye et al. 2022). Additionally, we introduce a task-recognition prediction, used exclusively during training, to help the model better differentiate between tasks.

Unified Modality Representation

To enable the one-stream transformer model to handle various SOT tasks, we convert the different modality inputs of each task into a unified token embedding form.

For the RGB-Depth, RGB-Thermal, and RGB-Event tracking tasks, the RGB data are paired with auxiliary modality data (depth, thermal, and event modalities, collectively referred to as DTE). Instead of converting the RGB and DTE modalities into separate token embeddings, we bind them together and jointly convert them using a proposed multi-modal patch embedding. This approach does not add significant computational overhead to the subsequent network. Specifically, the RGB image $\mathbf{I}_{\text{RGB}} \in \mathbb{R}^{H\times W\times 3}$ and the DTE image $\mathbf{I}_{\text{DTE}} \in \mathbb{R}^{H\times W\times 3}$ (noting that DTE data is stored as 3-channel images in current tracking datasets) are concatenated along the channel dimension, resulting in the concatenated image $\mathbf{I}_{\text{concat}} \in \mathbb{R}^{H\times W\times 6}$, summarized as follows:

$$\mathbf{I}_{\text{concat}} = \begin{bmatrix} \mathbf{I}_{\text{RGB}} \\ \mathbf{I}_{\text{DTE}} \end{bmatrix}. \tag{1}$$

Next, $\mathbf{I}_{\text{concat}}$ is divided into fixed-size patches, each with dimensions $P\times P\times 6$, where P is the patch size. Each $P\times P\times 6$ patch is then flattened into a one-dimensional vector of size $6P^2$. Finally, a linear transformation is applied to map the flattened patch vectors into an embedding space, as described by the following equation:

$$\mathbf{E}^{(i)} = \mathbf{W}_p\mathbf{P}^{(i)} + \mathbf{b}_p, \tag{2}$$

where $\mathbf{E}^{(i)}$ represents the embedding vector of the i-th patch with dimension D, $\mathbf{P}^{(i)}$ denotes the flattened vector of the i-th patch, $\mathbf{W}_p$ is the weight matrix of dimensions $D \times 6P^2$, and $\mathbf{b}_p$ is the bias term with dimension D. In this manner, the RGB-DTE data is transformed into a unified token embedding representation. For SOT tasks that do not include DTE data, such as RGB-based and RGB-Language tracking, we also use this multi-modal patch embedding by duplicating the RGB channels to create a 6-channel input.

For the language modality in RGB-Language tracking, we use a language model (CLIP-L (Radford et al. 2021) with an additional linear layer to adjust dimensions in our implementation) as the text encoder to extract a single-token feature embedding. This embedding is then concatenated with the multi-modal embeddings and fed into the transformer. For SOT tasks that do not include the language modality, we substitute with a fixed, nonsensical sentence.

Soft Token Type Embedding

LoRAT (Lin et al. 2024) proposes using token type embeddings to explicitly annotate type information for token embeddings, which enhances the distinction between the template foreground, template background, and the search region. However, token embeddings at the edges of the target bounding box often contain both foreground and background information, making it inaccurate to classify them solely as either template foreground or template background. To address this issue, this work introduces a soft token type embedding method to effectively account for both types in these cases.

Specifically, given a template image $\mathbf{I}^t_{\text{concat}} \in \mathbb{R}^{H \times W \times 6}$ with a bounding box $\mathbf{B}$ around the target, we first create a mask $\mathbf{M} \in \mathbb{R}^{H \times W}$ of the same size as the image. In this mask, the pixels inside the bounding box are assigned a value of 1, while pixels outside the bounding box are assigned a value of 0:

$$\mathbf{M}(i,j) = \begin{cases} 1 & \text{if } (i,j) \text{ is inside } \mathbf{B}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Next, we divide the mask $\mathbf{M}$ into non-overlapping patches of size $P \times P$. The k-th patch is denoted as $\mathbf{M}^{(k)}_{\text{patch}}$. For each patch, we compute the average value:

$$\mathbf{m}^{(k)}_{\text{avg}} = \frac{1}{P^2} \sum_{(i,j) \in \mathbf{M}^{(k)}_{\text{patch}}} \mathbf{M}(i,j), \tag{4}$$

where $\mathbf{m}^{(k)}_{\text{avg}}$ represents the average value of the k-th patch, indicating the degree to which the patch is considered foreground. Based on this average value, we enhance the multi-modal patch embeddings of the image with the corresponding token type embeddings. Specifically, for the k-th multi-modal patch embedding, we adjust it as follows:

$$\mathbf{E}^{(k)}_{\text{adj}} = \mathbf{E}^{(k)} + \mathbf{m}^{(k)}_{\text{avg}} \cdot \mathbf{E}_{\text{fg}} + (1 - \mathbf{m}^{(k)}_{\text{avg}}) \cdot \mathbf{E}_{\text{bg}}, \tag{5}$$

where $\mathbf{E}^{(k)}_{\text{adj}}$ denotes the adjusted embedding for the k-th patch, $\mathbf{E}^{(k)}$ represents the original multi-modal patch embedding, $\mathbf{E}_{\text{fg}}$ is the foreground token type embedding, and $\mathbf{E}_{\text{bg}}$ is the background token type embedding. This adjustment supplements the embeddings with more accurate foreground and background type information.

For the search region, where bounding box information is not available, we simply add a search region token type embedding $\mathbf{E}_{\text{search}}$ to each multi-modal patch embedding:

$$\mathbf{E}^{(k)}_{\text{adj}} = \mathbf{E}^{(k)} + \mathbf{E}_{\text{search}}. \tag{6}$$

Task-recognition Training Strategy

To enhance the model's ability to differentiate between various tasks, we introduce a task-recognition auxiliary training strategy. This approach explicitly teaches the model to identify the current task. Specifically, we compute the average of all feature embeddings output by the transformer model, resulting in a single vector $\mathbf{E}_{\text{avg}}$:

$$\mathbf{E}_{\text{avg}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{E}^{(i)}_{\text{output}}, \tag{7}$$

where N denotes the number of output token embeddings, and $\mathbf{E}^{(i)}_{\text{output}}$ represents the i-th output token embedding. This vector is then processed by a three-layer perceptron to classify the five tasks: RGB-based, RGB-Depth, RGB-Thermal, RGB-Event, and RGB-Language tracking:

$$\mathbf{y}_{\text{task}} = \text{MLP}(\mathbf{E}_{\text{avg}}), \tag{8}$$

where $\text{MLP}(\cdot)$ represents the three-layer perceptron used for task classification, and the output $\mathbf{y}_{\text{task}}$ represents the predicted probabilities for the five tasks. The predicted task probabilities $\mathbf{y}_{\text{task}}$ are then used to compute the cross-entropy loss against the true task labels $\mathbf{y}_{\text{true}}$:

$$\mathcal{L}_{\text{task}} = -\sum_{j=1}^{K} \mathbf{y}^{(j)}_{\text{true}} \log(\mathbf{y}^{(j)}_{\text{task}}), \tag{9}$$

where K denotes the number of tasks (5 in our case), $\mathbf{y}^{(j)}_{\text{true}}$ represents the ground truth label for task j, and $\mathbf{y}^{(j)}_{\text{task}}$ denotes the predicted probability for task j. Experimental results (see Section "Ablation and Analysis") demonstrate that this explicit task supervision enhances the model's performance. We note that this task-recognition strategy is used exclusively during training and does not impact the inference process.

Training and Inference

We train the model by mixing data from all five SOT tasks in each batch. This strategy allows the model to handle all five tasks after a single training phase. For tracking predictions, following OSTrack (Ye et al. 2022), we use a weighted focal loss for classification and a combination of ℓ_1 loss and generalized IoU loss for regression. For task-recognition predictions, we use the cross-entropy loss as described earlier. The overall loss function is summarized as:

$$\mathcal{L} = \mathcal{L}_{\text{class}} + \lambda_G \mathcal{L}_{\text{IoU}} + \lambda_{L_1} \mathcal{L}_{L_1} + \mathcal{L}_{\text{task}}, \tag{10}$$

where $\mathcal{L}_{\text{class}}$ denotes the weighted focal loss used for classification, $\mathcal{L}_{\text{IoU}}$ represents the generalized IoU loss, $\mathcal{L}_{L_1}$ is the

Model	Params (M)	FLOPs (G)	Speed (*fps*)
SUTrack-L384	247(+85)	223	12
SUTrack-L224	247(+85)	76	35
SUTrack-B384	70(+85)	67	32
SUTrack-B224	70(+85)	23	55
SUTrack-T224	22(+85)	6	100

Table 1: Details of SUTrack model variants.

ℓ_1 regression loss, $\mathcal{L}_{\text{task}}$ is the cross-entropy loss for task-recognition, and $\lambda_G = 2$ and $\lambda_{L_1} = 5$ are the regularization parameters. For inference, we adopt a conventional template update strategy (Yan et al. 2021a), employing two templates: one static and one updated dynamically during tracking. The template update mechanism is governed by a straightforward approach, utilizing a fixed interval and a confidence threshold to determine when updates occur.

Experiments

Implementation Details

The SUTrack models are implemented using Python 3.8 and PyTorch 1.11. Training is conducted on 4 NVIDIA A40 GPUs, while inference speed is evaluated on a single NVIDIA 2080TI GPU.

Model. We develop five variants of SUTrack models to strike a trade-off between speed and accuracy, each utilizing different transformer encoders and input resolutions. We adopt HiViT-L (Zhang et al. 2023) as the transformer encoder for SUTrack-L384 and L224, HiViT-B for SUTrack-B384 and B224, and HiViT-T for SUTrack-T224. The transformer encoders are initialized with the Fast-iTPN (Tian et al. 2024) pre-trained parameters. The numbers in the model names indicate the resolution of the search region. We present the model parameters, FLOPs, and inference speed in Tab. 1. For the parameters, a subscript of $+85$ denotes those specific to the text encoder CLIP-L, which can be omitted for tasks that do not involve language processing.

Training. Our training data comprises commonly used datasets for five SOT tasks, including COCO (Lin et al. 2014), LaSOT (Fan et al. 2021), GOT-10k (Huang, Zhao, and Huang 2019), TrackingNet (Muller et al. 2018), VASTTrack (Peng et al. 2024), DepthTrack (Yan et al. 2021c), VisEvent (Wang et al. 2024), LasHeR (Li et al. 2021), and TNL2K (Wang et al. 2021b). In each batch, we sample and mix data from these datasets, with RGB data being sampled at twice the rate of multi-modal data. The template and search images are generated by expanding the target bounding boxes by factors of 2 and 4, respectively. We train the model with AdamW optimizer. The model is trained for a total of 180 epochs, with $100,000$ image pairs per epoch.

Inference. The online template update interval is set to 25, with an update confidence threshold of 0.7 by default. A Hanning window penalty is applied to incorporate positional prior information in tracking, following standard practices (Chen et al. 2021; Ye et al. 2022).

State-of-the-Art Comparisons

We compare our SUTrack with state-of-the-art (SOTA) trackers spanning five tasks: RGB-based, RGB-Depth, RGB-Thermal, RGB-Event, and RGB-Language tracking. We note that SUTrack unifies these SOT tasks within a single model.

RGB-based Tracking. We evaluate our SUTrack on four large-scale RGB-based tracking benchmarks, including the long-term benchmarks LaSOT (Fan et al. 2021) and LaSOT$_{ext}$ (Fan et al. 2021), as well as the short-term benchmarks TrackingNet (Muller et al. 2018) and GOT-10k (Huang, Zhao, and Huang 2019). The results are presented in Tab. 2. Compared to trackers using the base model, our SUTrack-B224 surpasses all previous trackers across all four benchmarks, with the higher resolution SUTrack-B384 delivering even better results. Specifically, SUTrack-B384 achieves AUC scores of 74.4% on LaSOT, 86.5% on TrackingNet, and an AO score of 79.3% on GOT-10k, surpassing the previous best tracker, ODTrack (Zheng et al. 2024), by 1.2%, 1.4%, and 2.3% points, respectively. On LaSOT$_{ext}$, SUTrack-B224 matches the performance of the previous best tracker, LoRAT-B378, achieving an AUC score of 53.1%. When compared to trackers using large models, SUTrack-L384 and L224 also demonstrate competitive performance, setting new SOTA results on LaSOT, TrackingNet, and GOT-10k, while achieving the second-best performance on LaSOT$_{ext}$.

Efficient RGB-based Tracking. We develop the SUTrack-T224 model for edge devices with limited computational resources, and compare its performance with SOTA efficient trackers. The results are detailed in Tab. 3, which also includes the running speeds on both the Intel Core i9-9900K @ 3.60GHz CPU and the NVIDIA Jetson AGX Xavier edge device. Our method not only achieves real-time speeds on edge devices (with the real-time line defined as 20 *fps* by the VOT challenge (Kristan et al. 2020)) but also significantly outperforms previous efficient trackers. Specifically, SUTrack-T224 surpasses the previous best performances by 5%, 6.6%, 2.7%, and 7.6% on LaSOT, LaSOT$_{ext}$, TrackingNet, and GOT-10K, respectively.

RGB-Depth Tracking. As presented in Tab. 4, for the RGB-Depth tracking task, our SUTrack models set new state-of-the-art performance on both the VOT-RGBD22 (Kristan et al. 2023) and DepthTrack (Yan et al. 2021c) benchmarks. Specifically, on the VOT-RGBD22 benchmark, both SUTrack-L384 and SUTrack-B384 achieve an EAO score of 76.6%, surpassing the previous best, SDSTrack (Hou et al. 2024), by 3.8%. On the DepthTrack benchmark, SUTrack-L384 achieves an F-score of 66.4%, outperforming SDSTrack by 4.5%.

RGB-Thermal Tracking. As presented in Tab. 4, on the LasHeR (Li et al. 2021) benchmark, both SUTrack-L384 and SUTrack-L224 achieve an AUC score of 61.9%, surpassing the previous best, OneTracker, by 8.1% points. On the RGBT234 (Li et al. 2019b) benchmark, SUTrack-L224 obtains an AUC score of 70.8%, exceeding the performance of OneTracker by 6.6% points. These results highlight the significant performance advantage of our SUTrack model for RGB-Thermal tracking.

Method	LaSOT			LaSOT$_{ext}$			TrackingNet			GOT-10k		
	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P	AO	SR$_{0.5}$	SR$_{0.75}$
SUTrack-B384	**74.4**	**83.9**	**81.9**	52.9	63.6	60.1	**86.5**	**90.7**	**86.8**	**79.3**	**88.0**	**80.0**
SUTrack-B224	73.2	83.4	80.5	**53.1**	64.2	60.5	85.7	90.3	85.1	77.9	87.5	78.5
ODTrack-B384 (Zheng et al. 2024)	73.2	83.2	80.6	52.4	63.9	60.1	85.1	90.1	84.9	77.0	87.9	75.1
LoRAT-B378 (Lin et al. 2024)	72.9	81.9	79.1	**53.1**	**64.8**	**60.6**	84.2	88.4	83.0	73.7	82.6	72.9
ARTrackV2-256 (Bai et al. 2024)	71.6	80.2	77.2	50.8	61.9	57.7	84.9	89.3	84.5	75.9	85.4	72.7
AQATrack-256 (Xie et al. 2024)	71.4	81.9	78.6	51.2	62.2	58.9	83.8	88.6	83.1	73.8	83.2	72.1
OneTracker-384 (Hong et al. 2024)	70.5	79.9	76.5	-	-	-	83.7	88.4	82.7	-	-	-
EVPTrack-224 (Shi et al. 2024)	70.4	80.9	77.2	48.7	59.5	55.1	83.5	88.3	-	73.3	83.6	70.7
MixViT-288 (Cui et al. 2024)	69.6	79.9	75.9	-	-	-	83.5	88.3	83.5	72.5	82.4	69.9
Trackers with larger models												
SUTrack-L384	**75.2**	**84.9**	**83.2**	53.6	64.2	60.5	**87.7**	**91.7**	**88.7**	**81.5**	**89.5**	**83.3**
SUTrack-L224	73.5	83.3	80.9	54.0	65.3	61.7	86.5	90.9	86.7	81.0	90.4	82.4
LoRAT-L378 (Lin et al. 2024)	75.1	84.1	82.0	**56.6**	**69.0**	**65.1**	85.6	89.7	85.4	77.5	86.2	78.1
ODTrack-L384 (Zheng et al. 2024)	74.0	84.2	82.3	53.9	65.4	61.7	86.1	91.0	86.7	78.2	87.2	77.3
ARTrackV2-L384 (Bai et al. 2024)	73.6	82.8	81.1	53.4	63.7	60.2	86.1	90.4	86.2	79.5	87.8	79.6
ARTrack-L384 (Wei et al. 2023)	73.1	82.2	80.3	52.8	62.9	59.7	85.6	89.6	86.0	78.5	87.4	77.8
MixViT-L384 (Cui et al. 2024)	72.4	82.2	80.1	-	-	-	85.4	90.2	85.7	75.7	85.3	75.1
SeqTrack-L384 (Chen et al. 2023)	72.5	81.5	79.3	50.7	61.6	57.5	85.5	89.8	85.8	74.8	81.9	72.2

Table 2: State-of-the-art comparisons on four large-scale benchmarks. Methods employing the large model and the base model are compared separately. The number in the method name represents the resolution of the search region. The top two results are highlight with **bold** and underlined fonts, respectively.

Method	LaSOT			LaSOT$_{ext}$			TrackingNet			GOT-10k			Speed (*fps*)	
	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P	AUC	P$_{Norm}$	P	AO	SR$_{0.5}$	SR$_{0.75}$	CPU	AGX
SUTrack-T224	**69.6**	**79.3**	**75.4**	**50.2**	**61.1**	**57.0**	**82.7**	**87.2**	**80.8**	**72.7**	**82.1**	**70.5**	23	34
MixformerV2-S (Cui et al. 2023)	60.6	69.9	60.4	43.6	-	46.2	75.8	81.1	70.4	-	-	-	30	-
HiT (Kang et al. 2023)	64.6	73.3	68.1	-	-	-	80.0	84.4	77.3	64.0	72.1	58.1	33	61
FEAR-L (Borsuk et al. 2022)	57.9	-	60.9	-	-	-	-	-	-	64.5	74.6	-	-	-
HCAT (Chen et al. 2022b)	59.3	68.7	61.0	-	-	-	76.6	82.6	72.9	65.1	76.5	56.7	45	55
E.T.Track (Blatter et al. 2023)	59.1	-	-	-	-	-	75.0	80.3	70.6	-	-	-	47	20
LightTrack (Yan et al. 2021b)	53.8	-	53.7	-	-	-	72.5	77.8	69.5	61.1	71.0	-	41	36

Table 3: State-of-the-art comparisons of efficient tracking on four large-scale benchmarks.

RGB-Event Tracking. As presented in Tab. 4, SUTrack-L224, L384, and B384 secure the top three positions on the RGB-Event tracking benchmark, VisEvent (Wang et al. 2024). Notably, SUTrack-L224 attains the highest AUC score of 64.0%, surpassing the previous best, OneTracker, by 3.2 % points.

RGB-Language Tracking. As presented in Tab. 6, our five SUTrack models take the top five spots on the RGB-Language tracking benchmark, TNL2K (Wang et al. 2021b). SUTrack-L384 achieves the highest AUC score of 67.9%, significantly surpassing the previous best, OneTracker, by 9.9 % points. On the small-scale OTB99 (Li et al. 2017) benchmark, SUTrack demonstrates competitive performance despite not using the OTB99 training set, which other algorithms rely on.

Ablation and Analysis

The results of the ablation study are presented in Tab.5, where SUTrack-B224 serves as the baseline model, as shown in row #1.

Multi-Task v.s. Single-Task. In Tab.5 (#2), we train separate models for each SOT task. Compared to our multi-task unified model, single-task models show inferior performance across all tasks. The decline is particularly notable for RGB-Depth, RGB-Thermal, and RGB-Event tracking, where the training data is relatively small. This underscores the benefit of multi-task unification, which leverages shared knowledge across tasks to boost overall performance.

Zero-Shot Performance. In Tab. 5 (#3), we evaluate the zero-shot performance of the model by training it on all tasks' data except for the task being assessed. Although the results indicate a significant drop in performance, the model exhibits some zero-shot generalization capabilities. Notably, for specific tasks such as RGB-Depth and RGB-Thermal, the performance is comparable to or even exceeds that of the single-task models reported in #2.

Task-recognition Training Strategy. Tab. 5 (#4) presents the results after omitting the task-recognition auxiliary training strategy. This results in a decrease in performance compared to our default method. The potential reason for this is

Method	VOT-RGBD22			DepthTrack			LasHeR		RGBT234		VisEvent	
	EAO	Acc.	Rob.	F-score	Re	Pr	AUC	P	MSR	MPR	AUC	P
SUTrack-L384	**76.6**	83.5	**92.2**	**66.4**	**66.4**	**66.5**	**61.9**	76.9	70.3	93.7	63.8	80.5
SUTrack-L224	76.4	83.4	91.9	64.3	64.6	64.0	**61.9**	**77.0**	**70.8**	**94.6**	**64.0**	**80.9**
SUTrack-B384	**76.6**	**83.9**	91.4	64.4	64.2	64.6	60.9	75.8	69.2	92.1	63.4	79.8
SUTrack-B224	76.5	82.8	91.8	65.1	65.7	64.5	59.9	74.5	69.5	92.2	62.7	79.9
SUTrack-T224	68.1	81.0	83.9	61.7	62.1	61.2	53.9	66.7	63.8	85.9	58.8	75.7
OneTracker (Hong et al. 2024)	72.7	81.9	87.2	60.9	60.4	60.7	53.8	67.2	64.2	85.7	60.8	76.7
SDSTrack (Hou et al. 2024)	72.8	81.2	88.3	61.9	60.9	61.4	53.1	66.5	62.5	84.8	59.7	76.7
Un-Track (Wu et al. 2024)	72.1	82.0	86.9	61.0	60.8	61.1	-	-	62.5	84.2	58.9	75.5
ViPT (Zhu et al. 2023a)	72.1	81.5	87.1	59.4	59.6	59.2	52.5	65.1	61.7	83.5	59.2	75.8
ProTrack (Yang et al. 2022)	65.1	80.1	80.2	57.8	57.3	58.3	42.0	53.8	59.9	79.5	47.1	63.2

Table 4: SOTA comparisons on RGB-Depth, RGB-Thermal, and RGB-Event tracking.

#	Method	LaSOT	VOT-RGBD22	LasHeR	VisEvent	TNL2K	Δ
1	Baseline	73.2	76.5	59.9	62.7	65.0	–
2	Multi-Task → Single-Task	72.7	57.0	50.1	56.7	61.8	**-7.8**
3	Multi-Task → Zero-Shot	58.2	62.3	49.1	50.1	58.8	**-11.8**
4	W/o Task Recognition	72.6	76.5	59.8	62.5	63.9	**-0.4**
5	More RGB → Uniform	71.6	75.8	59.2	62.0	64.3	**-0.9**
6	Separate Representation	72.0	78.2	61.2	65.2	65.2	**+0.9**
7	W/o Token Type Embedding	72.4	76.2	59.4	61.5	64.6	**-0.6**
8	Soft → Hard	72.7	76.3	59.8	62.4	64.7	**-0.3**

Table 5: Ablation Study. Δ denotes the performance change (averaged over benchmarks) compared with the baseline.

Method	TNL2K		OTB99	
	AUC	P	AUC	P
SUTrack-L384	**67.9**	**72.1**	71.2	93.1
SUTrack-L224	66.7	70.3	72.7	94.4
SUTrack-B384	65.6	69.3	69.7	91.2
SUTrack-B224	65.0	67.9	70.8	93.4
SUTrack-T224	60.9	62.3	67.4	88.6
OneTracker (Hong et al. 2024)	58.0	59.1	69.7	91.5
JointNLT (Zhou et al. 2023)	56.9	58.1	65.3	85.6
DecoupleTNL(Ma and Wu 2023)	56.7	56.0	**73.8**	**94.8**
Zhao *et al.* (Zhao et al. 2023)	56.0	-	69.9	91.2
Li *et al.*(Li et al. 2022)	44.0	45.0	69.0	91.0
TNL2K-2 (Wang et al. 2021b)	42.0	42.0	68.0	88.0

Table 6: SOTA comparisons on RGB-Language tracking.

that explicit task supervision helps the model differentiate between data types, enabling it to better learn the specific characteristics of each task.

Data Ratio. In multi-task joint training, we sample RGB data at twice the rate of multi-modal data. The results of uniform sampling, as shown in Tab. 5 (#5), reveal a drop in performance. This is due to the limited diversity of multi-modal datasets, where an excessive proportion of such data can hinder model robustness. We look forward to the availability of larger-scale multi-modal datasets in the future.

Depth/Thermal/Event Modality Representation. We use multi-modal patch embedding to jointly represent RGB and Depth/Thermal/Event (DTE) modality image pairs. In Tab. 5 (#6), we explore an alternative, more computationally intensive method: applying standard patch embedding separately to RGB and DTE modalities, and then concatenating them along the spatial dimension. This approach yields higher performance, highlighting the potential of SUTrack. However, it results in nearly double the computational load. For efficiency, we have chosen to use our default multi-modal patch embedding method.

Token Type Embedding. In Tab. 5 (#7 and #8), we compare our soft token type embedding with results from both the absence of token type embedding and the original hard token type embedding method (Lin et al. 2024). Our soft token type embedding achieves superior performance by providing more precise token type information, which aids the model in distinguishing between template background, foreground, and search region tokens.

Conclusion

This work proposes a simple yet unified SOT framework, *i.e.*, SUTrack, which integrates five SOT tasks into a unified model trained in one session. SUTrack shows that a single model with a unified input representation is capable of managing diverse SOT tasks, eliminating the necessity for separate task-specific models or training processes. Extensive experiments demonstrate that SUTrack is effective, achieving competitive performance across all five SOT tasks. We hope SUTrack could serve as a solid foundation for future research on unified single object tracking.

Acknowledgments

The paper is supported in part by National Natural Science Foundation of China (nos. U23A20384, 62293540, 62293542), and in part by Talent Fund of Liaoning Province (no. XLYC2203014).

References

Bai, Y.; Zhao, Z.; Gong, Y.; and Wei, X. 2024. ARTrackV2: Prompting Autoregressive Tracker Where to Look and How to Describe. In *CVPR*.

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. S. 2016. Fully-Convolutional Siamese Networks for Object Tracking. In *ECCV*.

Bhat, G.; Danelljan, M.; Gool, L. V.; and Timofte, R. 2019. Learning Discriminative Model Prediction for Tracking. In *ICCV*.

Blatter, P.; Kanakis, M.; Danelljan, M.; and Van Gool, L. 2023. Efficient Visual Tracking with Exemplar Transformers. In *WACV*.

Borsuk, V.; Vei, R.; Kupyn, O.; Martyniuk, T.; Krashenyi, I.; and Matas, J. 2022. FEAR: Fast, Efficient, Accurate and Robust Visual Tracker. In *ECCV*.

Cai, Y.; Liu, J.; Tang, J.; and Wu, G. 2023. Robust Object Modeling for Visual Tracking. In *ICCV*.

Chen, B.; Li, P.; Bai, L.; Qiao, L.; Shen, Q.; Li, B.; Gan, W.; Wu, W.; and Ouyang, W. 2022a. Backbone is All Your Need: A Simplified Architecture for Visual Object Tracking. In *ECCV*.

Chen, X.; Kang, B.; Wang, D.; Li, D.; and Lu, H. 2022b. Efficient Visual Tracking via Hierarchical Cross-Attention Transformer. In *ECCVW*.

Chen, X.; Peng, H.; Wang, D.; Lu, H.; and Hu, H. 2023. SeqTrack: Sequence to Sequence Learning for Visual Object Tracking. In *CVPR*.

Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; and Lu, H. 2021. Transformer Tracking. In *CVPR*.

Cui, Y.; Jiang, C.; Wang, L.; and Wu, G. 2022. MixFormer: End-to-End Tracking with Iterative Mixed Attention. In *CVPR*.

Cui, Y.; Jiang, C.; Wang, L.; and Wu, G. 2024. MixFormer: End-to-End Tracking with Iterative Mixed Attention. *IEEE TPAMI*.

Cui, Y.; Song, T.; Wu, G.; and Wang, L. 2023. MixFormerV2: Efficient Fully Transformer Tracking. In *NeurIPS*.

Danelljan, M.; Bhat, G.; Khan, F. S.; and Felsberg, M. 2017. ECO: Efficient Convolution Operators for Tracking. In *CVPR*.

Danelljan, M.; Bhat, G.; Khan, F. S.; and Felsberg, M. 2019. ATOM: Accurate Tracking by Overlap Maximization. In *CVPR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Fan, H.; Bai, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Huang, M.; Liu, J.; Xu, Y.; et al. 2021. LaSOT: A High-Quality Large-Scale Single Object Tracking Benchmark. *IJCV*.

Feng, Q.; Ablavsky, V.; Bai, Q.; and Sclaroff, S. 2021. Siamese Natural Language Tracker: Tracking by Natural Language Descriptions with Siamese Trackers. In *CVPR*.

He, K.; Zhang, C.; Xie, S.; Li, Z.; and Wang, Z. 2023. Target-Aware Tracking with Long-Term Context Attention. In *AAAI*.

Hong, L.; Yan, S.; Zhang, R.; Li, W.; Zhou, X.; Guo, P.; Jiang, K.; Chen, Y.; Li, J.; Chen, Z.; and Zhang, W. 2024. OneTracker: Unifying Visual Object Tracking with Foundation Models and Efficient Tuning. In *CVPR*.

Hou, X.; Xing, J.; Qian, Y.; Guo, Y.; Xin, S.; Chen, J.; Tang, K.; Wang, M.; Jiang, Z.; Liu, L.; and Liu, Y. 2024. SDSTrack: Self-Distillation Symmetric Adapter Learning for Multi-Modal Visual Object Tracking. In *CVPR*.

Huang, L.; Zhao, X.; and Huang, K. 2019. GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild. *IEEE TPAMI*.

Kang, B.; Chen, X.; Wang, D.; Peng, H.; and Lu, H. 2023. Exploring Lightweight Hierarchical Vision Transformers for Efficient Visual Tracking. In *ICCV*.

Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Kämäräinen, J.-K.; Chang, H. J.; Danelljan, M.; Zajc, L. Č.; Lukežič, A.; et al. 2023. The Tenth Visual Object Tracking VOT2022 Challenge Results. In *ECCVW*. Springer.

Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Kämäräinen, J.-K.; Danelljan, M.; Zajc, L. Č.; Lukežič, A.; Drbohlav, O.; et al. 2020. The Eighth Visual Object Tracking VOT2020 Challenge Results. In *ECCV*.

Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; and Yan, J. 2019a. SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks. In *CVPR*.

Li, B.; Yan, J.; Wu, W.; Zhu, Z.; and Hu, X. 2018. High Performance Visual Tracking with Siamese Region Proposal Network. In *CVPR*.

Li, C.; Liang, X.; Lu, Y.; Zhao, N.; and Tang, J. 2019b. RGB-T Object Tracking: Benchmark and Baseline. *PR*.

Li, C.; Xue, W.; Jia, Y.; Qu, Z.; Luo, B.; Tang, J.; and Sun, D. 2021. LasHeR: A Large-Scale High-Diversity Benchmark for RGBT Tracking. *IEEE TIP*.

Li, Y.; Yu, J.; Cai, Z.; and Pan, Y. 2022. Cross-Modal Target Retrieval for Tracking by Natural Language. In *CVPR*.

Li, Z.; Tao, R.; Gavves, E.; Snoek, C. G.; and Smeulders, A. W. 2017. Tracking by Natural Language Specification. In *CVPR*.

Lin, L.; Fan, H.; Xu, Y.; and Ling, H. 2022. SwinTrack: A Simple and Strong Baseline for Transformer Tracking. In *NeurIPS*.

Lin, L.; Fan, H.; Zhang, Z.; Wang, Y.; Xu, Y.; and Ling, H. 2024. Tracking Meets LoRA: Faster Training, Larger Model, Stronger Performance. In *ECCV*.

Lin, T.-Y.; Maire, M.; Belongie, S. J.; Bourdev, L. D.; Girshick, R. B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*.

Ma, D.; and Wu, X. 2021. Capsule-based Object Tracking with Natural Language Specification. In *ACM MM*.

Ma, D.; and Wu, X. 2023. Tracking by Natural Language Specification with Long Short-term Context Decoupling. In *ICCV*.

Mayer, C.; Danelljan, M.; Bhat, G.; Paul, M.; Paudel, D. P.; Yu, F.; and Van Gool, L. 2022. Transforming Model Prediction for Tracking. In *CVPR*.

Mayer, C.; Danelljan, M.; Paudel, D. P.; and Van Gool, L. 2021. Learning Target Candidate Association to Keep Track of What not to Track. In *ICCV*.

Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; and Ghanem, B. 2018. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. In *ECCV*.

Nam, H.; and Han, B. 2016. Learning Multi-domain Convolutional Neural Networks for Visual Tracking. In *CVPR*.

Peng, L.; Gao, J.; Liu, X.; Li, W.; Dong, S.; Zhang, Z.; Fan, H.; and Zhang, L. 2024. VastTrack: Vast Category Visual Object Tracking. *arXiv preprint arXiv:2403.03493*.

Qian, Y.; Yan, S.; Lukežič, A.; Kristan, M.; Kämäräinen, J.-K.; and Matas, J. 2021. DAL: A Deep Depth-Aware Long-term Tracker. In *ICPR*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *ICML*.

Shi, L.; Zhong, B.; Liang, Q.; Li, N.; Zhang, S.; and Li, X. 2024. Explicit Visual Prompts for Visual Object Tracking. In *AAAI*.

Song, Z.; Luo, R.; Yu, J.; Chen, Y.-P. P.; and Yang, W. 2023. Compact Transformer Tracker with Correlative Masked Modeling. In *AAAI*.

Tao, R.; Gavves, E.; and Smeulders, A. W. M. 2016. Siamese Instance Search for Tracking. In *CVPR*.

Tian, Y.; Xie, L.; Qiu, J.; Jiao, J.; Wang, Y.; Tian, Q.; and Ye, Q. 2024. Fast-iTPN: Integrally Pre-trained Transformer Pyramid Network with Token Migration. *IEEE TPAMI*.

Wang, N.; Zhou, W.; Wang, J.; and Li, H. 2021a. Transformer Meets Tracker: Exploiting Temporal Context for Robust Visual Tracking. In *CVPR*.

Wang, X.; Li, J.; Zhu, L.; Zhang, Z.; Chen, Z.; Li, X.; Wang, Y.; Tian, Y.; and Wu, F. 2024. VisEvent: Reliable Object Tracking via Collaboration of Frame and Event Flows. *IEEE TCYB*.

Wang, X.; Shu, X.; Zhang, Z.; Jiang, B.; Wang, Y.; Tian, Y.; and Wu, F. 2021b. Towards More Flexible and Accurate Object Tracking with Natural Language: Algorithms and Benchmark. In *CVPR*.

Wei, X.; Bai, Y.; Zheng, Y.; Shi, D.; and Gong, Y. 2023. Autoregressive Visual Tracking. In *CVPR*.

Wu, Q.; Yang, T.; Liu, Z.; Wu, B.; Shan, Y.; and Chan, A. B. 2023. DropMAE: Masked Autoencoders with Spatial-Attention Dropout for Tracking Tasks. In *CVPR*.

Wu, Z.; Zheng, J.; Ren, X.; Vasluianu, F.-A.; Ma, C.; Paudel, D. P.; Van Gool, L.; and Timofte, R. 2024. Single-Model and Any-Modality for Video Object Tracking. In *CVPR*.

Xiao, Y.; Yang, M.; Li, C.; Liu, L.; and Tang, J. 2022. Attribute-based Progressive Fusion Network for RGBT Tracking. In *AAAI*.

Xie, F.; Chu, L.; Li, J.; Lu, Y.; and Ma, C. 2023. VideoTrack: Learning to Track Objects via Video Transformer. In *CVPR*.

Xie, F.; Wang, C.; Wang, G.; Cao, Y.; Yang, W.; and Zeng, W. 2022. Correlation-Aware Deep Tracking. In *CVPR*.

Xie, J.; Zhong, B.; Mo, Z.; Zhang, S.; Shi, L.; Song, S.; and Ji, R. 2024. Autoregressive Queries for Adaptive Tracking with Spatio-Temporal Transformers. In *CVPR*.

Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; and Yu, G. 2020. SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines. In *AAAI*.

Yan, B.; Peng, H.; Fu, J.; Wang, D.; and Lu, H. 2021a. Learning Spatio-Temporal Transformer for Visual Tracking. In *ICCV*.

Yan, B.; Peng, H.; Wu, K.; Wang, D.; Fu, J.; and Lu, H. 2021b. LightTrack: Finding Lightweight Neural Networks for Object Tracking via One-Shot Architecture Search. In *CVPR*.

Yan, S.; Yang, J.; Käpylä, J.; Zheng, F.; Leonardis, A.; and Kämäräinen, J.-K. 2021c. DepthTrack: Unveiling the power of RGBD tracking. In *ICCV*.

Yang, J.; Li, Z.; Zheng, F.; Leonardis, A.; and Song, J. 2022. Prompting for Multi-Modal Tracking. In *ACMMM*.

Ye, B.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2022. Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework. In *ECCV*.

Zhang, X.; Tian, Y.; Xie, L.; Huang, W.; Dai, Q.; Ye, Q.; and Tian, Q. 2023. HiViT: A Simpler and More Efficient Design of Hierarchical Vision Transformer. In *ICLR*.

Zhang, Z.; Peng, H.; Fu, J.; Li, B.; and Hu, W. 2020. Ocean: Object-Aware Anchor-Free Tracking. In *ECCV*.

Zhao, H.; Wang, X.; Wang, D.; Lu, H.; and Ruan, X. 2023. Transformer Vision-Language Tracking via Proxy Token Guided Cross-Modal Fusion. *PRL*.

Zheng, Y.; Zhong, B.; Liang, Q.; Mo, Z.; Zhang, S.; and Li, X. 2024. ODtrack: Online Dense Temporal Token Learning for Visual Tracking. In *AAAI*.

Zhou, L.; Zhou, Z.; Mao, K.; and He, Z. 2023. Joint Visual Grounding and Tracking with Natural Language Specification. In *CVPR*.

Zhu, J.; Lai, S.; Chen, X.; Wang, D.; and Lu, H. 2023a. Visual Prompt Multi-Modal Tracking. In *CVPR*.

Zhu, X.-F.; Xu, T.; Tang, Z.; Wu, Z.; Liu, H.; Yang, X.; Wu, X.-J.; and Kittler, J. 2023b. RGBD1K: A Large-Scale Dataset and Benchmark for RGB-D Object Tracking. In *AAAI*.