Efficient Motion Prompt Learning for Robust Visual Tracking

Jie Zhao¹ Xin Chen² Yongsheng Yuan¹ Michael Felsberg³ Dong Wang¹ Huchuan Lu¹

Abstract

Due to the challenges of processing temporal information, most trackers depend solely on visual discriminability and overlook the unique temporal coherence of video data. In this paper, we propose a lightweight and plug-and-play motion prompt tracking method. It can be easily integrated into existing vision-based trackers to build a joint tracking framework leveraging both motion and vision cues, thereby achieving robust tracking through efficient prompt learning. A motion encoder with three different positional encodings is proposed to encode the long-term motion trajectory into the visual embedding space, while a fusion decoder and an adaptive weight mechanism are designed to dynamically fuse visual and motion features. We integrate our motion module into three different trackers with five models in total. Experiments on seven challenging tracking benchmarks demonstrate that the proposed motion module significantly improves the robustness of vision-based trackers, with minimal training costs and negligible speed sacrifice. Code is available at https://github.com/ zj5559/Motion-Prompt-Tracking.

1. Introduction

Given a sequence and an arbitrary object with interest, visual object tracking (VOT) aims to locate this specified object in each subsequent frame. As one of the fundamental computer vision tasks, VOT has been developed rapidly over the past decade. Mainstream trackers (Li et al., 2019; Bhat et al., 2019; Zhao et al., 2022b; Chen et al., 2023) perceive VOT as a visual matching problem between a pair of discrete image patches, including a template and a local search image cropped from the initial and current frames, respectively. These vision-based tracking frameworks mainly rely



Figure 1. **Illustration of different tracking paradigms.** Our plugand-play method enables visual trackers to benefit from motion prompts, making them more akin to the human tracking paradigm.

on the discriminative ability of visual models but overlook the crucial temporal coherence in videos. The temporal consistency of these trackers can be only reflected in determining the next local search region by the predicted target position. As shown in Fig. 1(a), due to inherent limitations of appearance cues, these vision-based trackers are prone to tracking drift when confronted with complex challenges, like distractors, severe occlusion, and so on. Unlike relying solely on visual information, the human tracking paradigm (Ramachandran, 1985; Sokhandan & Monadjemi, 2024) seamlessly integrates both vision and motion cues. As shown in Fig. 1(b), humans can effortlessly identify the specified target among multiple visually similar objects by discerning its motion patterns from trajectories.

However, due to the blend of object and camera movement, the irregularity in observed motion patterns makes it challenging to incorporate temporal coherence into tracking frameworks. Recent state-of-the-art (SOTA) trackers (Wei et al., 2023; Bai et al., 2024; Zheng et al., 2024) start exploring effective temporal mechanisms to enhance tracking accuracy, but sequential training (Kim et al., 2022) is required for these methods, which considerably increases computational and memory demands. Beyond the substantial training costs, constrained computing resources will also limit the ability of these models to perceive long temporal information. In this paper, we post a question: Is heavy sequential training necessary for trackers to capture temporal relations? We answer this question by proposing a trainingefficient Motion Prompt-based Tracking module (MPT), which demonstrates that sequential training is unnecessary for temporal-related tracking.

¹Dalian University of Technology ²City University of Hong Kong ³Linköping University. Correspondence to: Dong Wang <wdice@dlut.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

Specifically, our MPT module can be flexibly integrated into existing vision-based trackers to achieve joint tracking based on both visual and motion cues. Within a lightweight framelevel fine-tuning, our method can enhance various baseline trackers to achieve comparable performance to those sequential-based SOTA methods. As shown in Fig. 1(c), our joint tracking framework contains a fixed vision-based tracker and several additionally proposed motion modules. The prompt encoder (the orange block) independently encodes historical trajectories into the visual embedding space by three different positional encodings. A fusion decoder (the green block) is then employed to dynamically fuse visual and motion features by a Transformer network and an adaptive weighting mechanism. Furthermore, we adopt prompt learning to integrate our motion modules into visionbased SOTA trackers more efficiently and flexibly. We develop our MPT as a plug-and-play prompt module, aiming to learn the ability for dynamic adaptation to baseline visual trackers. All parameters of the baseline are fixed during training, enabling our method to have fast training with minimal memory requirements. By integrating motion prompts from object trajectories, our MPT can sustain robust tracking under challenging scenarios, reflecting a tracking paradigm more akin to human-like behavior.

Our main contributions can be summarized as follows:

- We propose a plug-and-play motion prompt tracking module (MPT), which can be flexibly integrated into various tracking models to capture and utilize temporal coherence from historical trajectories.
- Efficient prompt learning is employed to fine-tune our method, which unbinds the necessity of heavy sequential training for temporal learning, thereby releasing huge requirements of training resources.
- We integrate our MPT into three visual trackers across five models. Experiments on seven benchmarks demonstrate that our MPT enhances the robustness of existing vision-based trackers in challenging scenarios, within minimal training costs and negligible speed sacrifice.

2. Related Work

In this section, we review mainstream visual trackers, temporal-related trackers, and prompt learning, respectively.

Visual Tracking. Existing trackers commonly adopt a visual matching strategy to model the tracking task. Most earlier trackers (Bertinetto et al., 2016; Dai et al., 2019; Li et al., 2019) employ correlation operations to compute the similarity map between visual features of the template and search region. TransT (Chen et al., 2021) and subsequent Transformer-based works (Ye et al., 2022; Lin et al., 2022; Chen et al., 2023) employ Transformer networks (Vaswani

et al., 2017; Dosovitskiy et al., 2021) to obtain the fused features of the template and search region, which is then used to make the final prediction. Due to global correlation and deep fusion, these trackers lead to improved accuracy compared to similarity-based methods. Despite achieving superior performance on existing benchmarks, such solely vision-based matching strategies encounter difficulties in complex visual environments. To alleviate this bottleneck, our work proposes a general tracking module based on motion prompts. By incorporating temporal motion cues into these vision-based trackers, we enrich the information available to the head network for predictions, thereby significantly improving the tracking robustness in challenging scenarios.

Temporal-related Tracking. Beyond visual information, some works have investigated the significance of temporal information for robust tracking. Some traditional trackers (Weng et al., 2006; Zhang et al., 2014) primarily utilize kalman filtering to predict the next motion step of the target. However, they assume that the target undergoes regular motion patterns, which usually do not align with the complex tracking scenarios. The lack of deep visual representations also leads to a substantial lag in their performance.

In the era of deep learning, some methods (Bhat et al., 2019; Zhao et al., 2022a; Yan et al., 2021; Cui et al., 2022; Liu et al., 2024) employ update mechanisms for models or templates to capture the temporal changes of visual information, these methods primarily depend on richer appearance information to enhance tracking. TCTrack++ (Cao et al., 2023) and ODTrack (Zheng et al., 2024) employ heavy temporal mechanisms, which implicitly integrate temporal information via propagating consecutive visual features. Different from them, our lightweight module directly encodes and integrates motion trajectories into the visual tracking framework, offering a fresh perspective. Our experiments with the template update tracker SeqTrack (Chen et al., 2023) also demonstrate that benefits derived from motion and updates are complementary rather than conflicting.

Recently, Alireza *et al.* (Sokhandan & Monadjemi, 2024) adopt optical flows to represent motion vectors, which raises computational burdens. SLT (Kim et al., 2022) proposes a sequential training method, incorporating motion cues into training by locating short video clips. But it still follows vision-based tracking in inference. Implicitly learning temporal correlations among consecutive frames is also a complex and computationally intensive task. ARTrack (Wei et al., 2023) and its extension (Bai et al., 2024) share a similar motivation with ours, integrating temporal information into tracking by individually encoding the motion trajectory. However, its motion encoding is tailored for autoregressive models, which limits its applicability to other tracking frameworks. Autoregressive model and sequential train-



Figure 2. **Pipeline of our motion prompt-based tracking method.** The historical trajectory is first embedded into the visual embedding space by our motion encoder, and then fused with the visual embedding using the proposed fusion decoder. An adaptive weight mechanism is employed to further dynamically adjust vision and motion cues. Ultimately, the obtained fused embedding is used to make a robust tracking prediction by the tracking head Head_{Tr} .

ing also increase memory and computational costs. Limited hardware memory will further restrict the model from perceiving long trajectories. In comparison, our method employs a low-cost prompt learning approach to leverage longer motion trajectories for guiding existing trackers, offering greater flexibility and versatility.

Prompt Learning. Prompt learning is an emerging machine learning paradigm, garnering particular attention in the field of natural language processing (Liu et al., 2023b). Its core idea involves interacting with the model via task-related prompts rather than directly adjusting model parameters. In computer vision and multi-modal domains, prompt learning proves to be more efficient and flexible than traditional fine-tuning methods. It shows success in various tasks, including visual recognition (Jia et al., 2022), vision-language understanding (Liu et al., 2023a), dense prediction (Chen et al., 2022b), and video understanding (Ju et al., 2022).

In the field of visual tracking, in addition to the aforementioned ARTrack and its extension, which adopt trajectories as prompts by employing sequential training and autoregressive models, ProTrack (Yang et al., 2022) and ViPT (Zhu et al., 2023) utilize prompt learning to achieve superior performance for multi-modal tracking, yet they do not exploit the potential of prompt learning in general visual tracking. PromptVT (Zhang et al., 2024) can be treated as an appearance-based prompt method. It efficiently enhances appearance features using dynamic appearance information, thereby enabling the model to be robust to appearance changes during tracking. However, the tracking problem is still treated as a discrete image processing task. In contrast, our MPT leverages the temporal consistency of videos by incorporating continuous object trajectories into the visual tracking framework, offering a new perspective on employing prompt learning in the tracking field. Our method proves to be more effective in addressing visually challenging scenarios, such as occlusions and distractors.

3. Method

3.1. Revisit of Vision-based Tracking Framework

Mainstream trackers formulate the tracking problem as

$$B^{t} = \operatorname{Head}_{\operatorname{Tr}}(\varphi_{v}(Z, X^{t})).$$
(1)

A visual encoder $\varphi_v(\cdot)$ is commonly employed to embed both the template image Z and the search image X^t , producing the discriminative visual representations, like similarity maps (Li et al., 2019; Bhat et al., 2019) or fused features (Cui et al., 2022; Chen et al., 2022a) of image pairs. The final tracking result B^t for the t-th frame is predicted by a tracking head module, Head_{Tr}. Some trackers (Ye et al., 2022; Yan et al., 2021) employ a lightweight head to directly predict coordinates, whereas other methods (Chen et al., 2023; Wei et al., 2023) implement the head module as a heavy decoder to tackle the sequential coordinate prediction problem. For these vision-based tracking methods, a powerful visual encoder φ_v is necessary to achieve robust tracking. However, good visual models need to achieve invariant representations to appearance changes for the same object, while maintaining distinctive for distractors. Acquiring such balanced discriminative visual representations poses a challenging learning task for visual models. In addition, when faced with occlusion, illumination changes, or distractors, visual information often turns out to be unreliable, leading to tracking drift in these vision-based trackers.

In this paper, we propose a motion prompt-based tracking method, which can integrate motion cues into vision-based baseline trackers by prompt learning. This joint tracking framework incorporating both motion and vision cues can adaptively extract complementary features, thereby enhancing the robustness of tracking predictions, particularly in the face of complex challenges.

3.2. Overview of Motion Prompt Tracking

Our tracking method with motion prompts is formulated as

$$B^{t} = \operatorname{Head}_{\operatorname{Tr}}(\boldsymbol{\mathcal{D}}_{f}(\varphi_{v}(Z, X^{t}), \boldsymbol{\varphi}_{m}(B^{t-T:t-1}))). \quad (2)$$

In addition to the visual encoder $\varphi_v(\cdot)$, we employ a motion encoder $\varphi_m(\cdot)$ to embed the historical trajectory $B^{t-T:t-1}$ into the latent embedding space consistent with visual features. The trajectory is denoted as consecutive bounding boxes of the target from previous T frames. A decoder $\mathcal{D}_f(\cdot)$ is then utilized to fuse motion and visual features, and the resulting fused representation will be forwarded to the downstream head to generate robust tracking predictions.

As shown in Fig. 2, the two input flows, including the visual input and the object motion trajectory, are processed separately through the baseline visual model and our motion encoder to extract corresponding features. Through calculations of several self-attention and cross-attention modules, features from the two branches are complementarily fused. An adaptive weight mechanism is additionally utilized to establish a dynamic residual connection between visual features and the output of the fusion decoder, thereby achieving stable fused embeddings. By integrating complementary motion prompts, the proposed motion modules empower the vision-based baseline tracker to achieve enhanced performance, especially under challenging scenes.

3.3. Model Architecture

Our motion prompt tracking method mainly consists of three modules, *i.e.*, a motion encoder, a fusion decoder, and an adaptive weight mechanism. Detailed architectures of each module are described as follows.

Motion Encoder. Given consecutive object bounding boxes of previous T frames, our motion encoder embeds these sequential coordinates into multiple motion tokens aligned with the visual embedding space. The following three types of positional encodings (PE) are utilized to characterize each token across both spatial and temporal dimensions.

(1) Spatial positional encoding. Inspired by SAM (Kirillov et al., 2023), we represent each bounding box using the normalized coordinates of two corner points, namely, the top-left and bottom-right points. A Gaussian-based spatial PE (Tancik et al., 2020) is adopted to map 2-dimensional point coordinates into C-dimensional spatial vectors, aligning the motion embedding with the visual embedding in terms of dimensions.

(2) Point embedding. After being mapped by the spatial PE, the trajectory of T frames is transformed to a group of motion tokens $\in \mathbb{R}^{2T \times C}$. To distinguish between the two corner point types for each token, we introduce two learnable point embeddings and add them to their respective motion tokens. This enables the model to perceive the point type of each motion token.

(3) Temporal positional encoding. To establish temporal orders for motion tokens, we incorporate a learnable temporal PE into each token. Tokens from the same frame share identical temporal PE. Since the motion information closer to the current frame is typically more crucial in a long-term trajectory, we initialize this temporal PE in a non-linear manner, formulated as follows:

$$PE(t,2i) = \sin\left(\frac{\alpha \lg(t+1)}{n^{2i/d}}\right),$$

$$PE(t,2i+1) = \cos\left(\frac{\alpha \lg(t+1)}{n^{2i/d}}\right).$$
(3)

Here, i, d denotes the dimension variable and the total dimension, $t \in [0, T - 1]$ represents the temporal position of the point in the trajectory. A larger t corresponds to a closer position to the current frame. The constant n is set to 10^4 , and α is a parameter employed to control the frequency range. To maximize spatial resolution without causing aliasing, we set $\alpha = 7.23$ according to the Nyquist frequency (Nyquist, 1928). Both theoretical proof and experimental analysis can be found in Appendix A.

In addition to the 2T motion tokens, we also introduce three learnable representative (Rep.) tokens. Among them, RT_{tl} and RT_{br} are employed to summarize motion information from all top-left and bottom-right trajectory points, respectively. RT_w represents the confidence of the motion prompt. We set two sub-tasks to guide the learning of three Rep. tokens, namely, bounding box regression and weight regression. These sub-tasks encourage our model to develop the ability to extract crucial motion prompts and predict dynamic weights. As the output of the motion encoder, we concatenate 2T motion tokens and the three Rep. tokens to represent the extracted motion feature.

Fusion Decoder. As shown in Fig. 2, we propose a lightweight fusion decoder to fuse the visual feature $f_v \in \mathbb{R}^{b \times HW \times C}$ and motion feature $f_m \in \mathbb{R}^{b \times (2T+3) \times C}$, where b denotes the batch size, and H, W represent resolutions of

the visual feature. The proposed fusion decoder consists of N Transformer-based blocks, each including a motion self-attention module, two cross-attention modules with different directions, and a multi-layer perception (MLP) layer. Specifically, the self-attention module first explores crucial motion information from motion features themselves. The first cross-attention module then fuses visual cues into motion features, in which motion features f_m are set as queries (q). After the update by an MLP, we take the three Rep. tokens as keys (k) and values (v) for the second cross-attention module, and update the visual feature f_v (as queries) with representative motion prompts. For each cross-attention module, we incorporate a dense spatial map as positional encoding onto visual features. This ensures a correspondence between visual and motion features in spatial coordinates. Regarding temporal positions, we also apply the temporal PE to motion features in the first cross-attention module.

Adaptive Weight Mechanism. Considering the instability of motion trajectories, we calculate the weighted average of the fused feature and original visual feature using an adaptive weight mechanism. A weight head Head_W is employed to dynamically predict the weight W_M according to the output RT_w from MLP. To explicitly guide the learning of W_M , we additionally introduce a motion head Head_M to predict the current bounding box B_M according to RT_{tl} and RT_{br} . We calculate the Intersection-over-Union (IoU) of the groundtruth and B_M , and use it to explicitly supervise the learning for W_M .

3.4. Training and Inference

Training. We adopt prompt learning to achieve efficient training. All parameters in the baseline tracker are fixed, and only our MPT modules are fine-tuned, including the motion encoder, fusion decoder, learnable embeddings, $Head_W$, and $Head_M$. The backpropagation process is illustrated as dashed red arrows in Fig. 2, The loss function for our MPT comprises the following three terms:

- Baseline tracking loss. We follow the same tracking loss as baseline trackers, namely $\mathcal{L}_{Tr}(B, GT_t)$, where B denotes the output of Head_{Tr}, and GT_t represents the groundtruth of the object bounding box.
- Motion regression loss. To enable Rep. tokens RT_{tl} and RT_{br} to extract essential motion patterns, we set a bounding box regression task on Head_M. The generalized IoU loss (Rezatofighi et al., 2019) and ℓ_1 loss are employed to supervise this sub-task, formulated as

$$\mathcal{L}_{\mathrm{M}} = \lambda_{\mathrm{IoU}} \mathcal{L}_{\mathrm{IoU}} + \lambda_{\ell_1} \mathcal{L}_1, \qquad (4)$$

where $\lambda_{IoU} = 2$ and $\lambda_{\ell_1} = 5$ in our experiments.

• Adaptive weight loss. In our assumption, a lower quality of the motion tracking prediction $B_{\rm M}$ typically im-

plies less confidence in motion prompts. We thereby adopt the IoU between B_M and GT_t as the label of the predicted weight W_M , and calculate MSE loss to achieve explicit supervision. The adaptive weight loss is formulated as

$$\mathcal{L}_{\mathrm{W}} = \mathcal{L}_{\mathrm{MSE}}(W_M, \mathrm{IoU}(B_{\mathrm{M}}, \mathrm{GT}_t)).$$
(5)

The overall loss function of our MPT involves the above three terms, represented as

$$\mathcal{L} = \mathcal{L}_{\mathrm{Tr}} + \lambda_{\mathrm{M}} (\mathcal{L}_{\mathrm{M}} + \mathcal{L}_{\mathrm{W}}), \tag{6}$$

where $\lambda_{\rm M} = 1$ in our experiments.

Inference. During the inference, predicted bounding boxes of previous T frames are first aligned and normalized based on the coordinate system of the current search region. Normalized trajectory is sent together with the image pair into our motion prompt-based tracking model for tracking prediction. As Head_M primarily serves to facilitate the more effective training for our motion modules, this network is disregarded during inference to save computational costs.

4. Experiment

To demonstrate the effectiveness and training efficiency of our MPT, we integrate our MPT into two different visual trackers, *i.e.*, a one-stream tracker OSTrack¹, and an autoregressive-based tracker SeqTrack. We also apply our MPT to ARTrack to show that our MPT can further complement temporal-related trackers rather than conflicting. We select a total of five baseline models with different backbones and resolutions. To distinguish them, base / large backbones are represented as "B / L", while the following numbers represent image resolutions. We compare our methods with their baselines and other SOTA trackers on seven challenging tracking benchmarks. Experimental results demonstrate that our MPT can be seamlessly integrated into various trackers, enhancing their tracking robustness with fast and memory-efficient training.

4.1. Implementation Details

Our methods are implemented in Python with PyTorch. Models are trained on 2 NVIDIA A100 GPUs, and tested on a single NVIDIA RTX2080Ti GPU.

Model. The length of the historical trajectory T is set to 30 based on experimental results. Each 2-dimensional coordinate is encoded into a motion token with the same dimension as baseline visual features. The lightweight fusion decoder is implemented as a two-layer Transformer network. The weight head Head_W and motion head Head_M are implemented by a two-layer MLP, where the hidden size is 256.

¹The official model without the candidate elimination module.

Mathad	VOT	2018	VOT2	2020	VOT2022 (STB)		
Method	EAO (†)	$R(\downarrow)$	EAO (†)	$R(\uparrow)$	EAO (\uparrow)	R (†)	
MixFormer-22k	0.228	1.836	0.317	0.824	0.533	0.844	
ROMTrack-B256	0.357	1.010	0.318	0.815	0.554	0.825	
OSTrack-B256	0.287	1.204	0.311	0.803	0.530	0.823	
+MPT (Ours)	0.406	0.473	0.334	0.846	0.572	0.859	
SeqTrack-B256	0.339	1.067	0.316	0.806	0.523	0.815	
+MPT (Ours)	0.428	0.471	0.315	0.810	0.528	0.825	
ARTrack-B256	0.399	0.599	0.315	0.809	0.532	0.818	
+MPT (Ours)	0.469	0.491	0.336	0.828	0.566	0.842	
MixFormer-L	0.238	1.596	0.325	0.825	0.549	0.843	
ROMTrack-B384	0.297	1.231	0.309	0.794	0.540	0.812	
ARTrack-L384	0.391	0.520	0.336	0.834	0.570	0.852	
OSTrack-B384	0.271	1.408	0.288	0.767	0.518	0.796	
+MPT (Ours)	0.363	0.679	0.314	0.819	0.548	0.826	
SeqTrack-L384	0.319	1.117	0.337	0.859	0.568	0.869	
+MPT (Ours)	0.461	0.414	0.341	0.870	0.579	0.882	
Average gain	+10.2%	+57.3%	+1.5%	+2.6%	+2.4%	+2.3%	

Table 1. **Performance comparison on three VOT datasets.** The best results are highlighted with **bold**.

Training. We adopt prompt learning to efficiently train our MPT, where the baseline model is frozen. We select the training splits of LaSOT (Fan et al., 2019), GOT-10K (Huang et al., 2019)², and TrackingNet (Muller et al., 2018) as the training data. For the motion input, we adopt DiMP-18 (Bhat et al., 2019) to generate real tracking predictions for each of training sequences, and employ reverse sampling, sparse sampling and CutMix (Yun et al., 2019) for data augmentations. More implementation details can be found in Appendix B, and extended experimental analysis in terms of data augmentations and trajectory length can be found in Appendix C.

4.2. State-of-the-Art Comparison

We compare our methods with baselines and other SOTA trackers on the following seven tracking benchmarks.

VOT. The committee of VOT challenge proposes a series of challenging tracking benchmarks. Trackers are evaluated primarily by the expected average overlap (EAO), which is a principled combination in terms of tracking accuracy and robustness (R). As shown in Tab. 1, we compare our methods with several SOTA trackers on three representative VOT datasets, including VOT2018 (Kristan et al., 2018), VOT2020 (Kristan et al., 2020), and VOT2022 (Kristan et al., 2022). For a fair comparison, all of the presented trackers are run with their official models and settings, and evaluated based on bounding box outputs. Compared with SOTA methods, trackers integrating our MPT exhibit superior performance on these challenging sequences, especially in terms of robustness. Our MPT significantly enhances the performance of the five baseline models, averaging an improvement of 2.0% EAO across VOT2020 and VOT2022, and 10.2% EAO on VOT2018.

LaSOT and LaSOT_{EXT}. LaSOT (Fan et al., 2019) contains 280 long-term sequences with 70 different categories, while its extension LaSOT_{EXT} (Fan et al., 2021), consists of 150 extremely challenging sequences with 15 unseen categories. The success rate (AUC), normalized precision (P_{norm}), and precision (Pre) are adopted to evaluate trackers. As reported in Tab. 2, our MPT increases baseline trackers by an average 1.6% AUC on LaSOT_{EXT}. Furthermore, our training-efficient method (+**MPT**) enhances SeqTrack-L384 to achieve an AUC of 73.9% on LaSOT, which is competitive with sequential-level training method using multi-template strategy, ODTrack (Zheng et al., 2024).

TNL2K (Wang et al., 2021). TNL2K is a recently released dataset with 700 challenging sequences. As shown in Tab. 2, our MPT yields average 0.6% and 1.0% improvements in terms of AUC and P_{norm} , respectively.

TrackingNet (Muller et al., 2018). TrackingNet contains 511 sequences covering diverse categories and scenarios. As presented in Tab. 2, our MPT brings baselines an average performance improvement of around 0.4% on each metric.

Discussion. We find that the performance improvement of our MPT is slightly biased across different datasets, with more substantial gains on more challenging ones (like VOTs, $LaSOT_{EXT}$). To explore the underlying reasons, we divide each of the three datasets (*i.e.*, LaSOT, LaSOT_{EXT}, and TNL2K) into "hard / easy" subsets based on the performance of each sequence, and analyze the performance changes of MPT on each subset (shown as Tab. 7). As discussed in Appendix C, our MPT shows great consistency in hard / easy sets, achieving average AUC changes of +4.4% / -0.8%, with small standard deviations of 0.6 / 0.4 across five baselines. Therefore, we reach a consistent conclusion that the advantage of our MPT is more evident in hard scenarios. This is because our MPT primarily aims to reduce tracking failures by motion cues, leading to a significant improvement in tracking robustness rather than accuracy. This can also be intuitively demonstrated by the performance gains on the three VOT benchmarks (shown as Tab. 1).

In addition, compared with some trackers like ODTrack and ARTrackV2, which adopt sequential-level training or multitemplate strategies to enhance their tracking performance, our MPT requires less memory and computing resources during training. As shown in Tab. 3, taking a sequentialtraining tracker (ARTrack-B256) as a reference, the setting of 8 batch sizes (BS) needs 50G GPU memory, within more than 3 days of training. In contrast, our MPT adopts lightweight architecture and efficient frame-level training, introducing only 13M parameters (Params) and negligible speed (FPS) degradation for baselines. For OSTrack with 128 BS, our MPT only needs 5 hours of fine-tuning using

²Following the VOT protocol, 1k sequences are removed.

Mathad	Train	т		LaSOT		L	aSOT _{EX}	Т		TNL2K		Ti	rackingN	let
Method	Irain	Temp.	AUC	$\mathrm{P}_{\mathrm{norm}}$	Pre	AUC	$\mathrm{P}_{\mathrm{norm}}$	Pre	AUC	$\mathrm{P}_{\mathrm{norm}}$	Pre	AUC	$\mathbf{P}_{\mathrm{norm}}$	Pre
SiamRCNN (Voigtlaender et al., 2020)	Frame	1	64.8	72.2	-	-	-	-	-	-	-	81.2	85.4	80.0
TransT (Chen et al., 2021)	Frame	1	64.9	73.8	69.0	-	-	-	50.7	-	51.7	81.4	86.7	80.3
STARK-ST101 (Yan et al., 2021)	Frame	2	67.1	77.0	-	-	-	-	-	-	-	82.0	86.9	-
ToMP-101 (Mayer et al., 2022)	Frame	1	68.5	79.2	73.5	45.9	-	-	-	-	-	81.5	86.4	78.9
AiATrack (Gao et al., 2022)	Frame	2	69.0	79.4	73.8	46.8	54.4	54.2	-	-	-	82.7	87.8	80.4
MixFormer-22k (Cui et al., 2022)	Frame	2	69.2	68.7	74.7	-	-	-	-	-	-	83.1	88.1	81.6
Sim-B16 (Chen et al., 2022a)	Frame	1	69.3	78.5	-	-	-	-	54.8	-	53.8	82.3	86.5	-
ROMTrack-B256 (Chen et al., 2022a)	Frame	2	69.3	78.8	75.6	48.9	59.3	55.0	-	-	-	83.6	88.4	82.7
MixViT (ConvMAE) (Cui et al., 2024)	Frame	2	70.4	80.4	<u>76.7</u>	-	-	-	-	-	-	84.5	89.1	83.7
OSTrack-B256 (Ye et al., 2022)	Frame	1	68.6	78.0	74.5	46.9	56.9	52.8	55.6	72.1	56.1	83.0	87.7	81.7
+MPT (Ours)	Frame	1	68.9	78.6	74.8	48.7	59.1	55.2	55.8	72.7	56.5	83.6	88.4	82.4
SeqTrack-B256 (Chen et al., 2023)	Frame	2	69.4	79.2	75.8	49.8	<u>61.2</u>	56.8	56.5	74.2	58.6	83.2	88.3	82.3
+MPT (Ours)	Frame	2	70.1	80.4	<u>76.7</u>	50.8	62.4	58.0	57.8	75.9	60.3	83.6	88.8	82.6
ARTrack-B256 (Wei et al., 2023)	Seq	1	70.4	79.5	76.6	48.4	57.7	53.7	57.9	74.0	59.6	84.2	88.7	83.5
+MPT (Ours)	Frame	1	70.3	80.1	76.8	<u>50.4</u>	60.7	<u>57.5</u>	58.5	<u>75.2</u>	60.9	84.2	88.9	83.2
ARTrackV2-B256 (Bai et al., 2024)	Seq	2	71.6	80.2	77.2	50.8	61.9	57.7	59.2	-	-	84.9	89.3	84.5
MixFormer-L (Cui et al., 2022)	Frame	2	70.1	79.9	76.3	-	-	-	-	-	-	83.9	88.9	83.1
Sim-L14 (Chen et al., 2022a)	Frame	1	70.5	79.7	-	-	-	-	55.6	-	55.7	83.4	87.4	-
ROMTrack-B384 (Cai et al., 2023)	Frame	2	71.4	81.4	78.2	51.3	62.4	58.6	-	-	-	84.1	89.0	83.7
MixViT-L (ConvMAE) (Cui et al., 2024)	Frame	2	<u>73.3</u>	<u>82.8</u>	<u>80.3</u>	-	-	-	-	-	-	86.1	90.3	86.0
OSTrack-B384 (Ye et al., 2022)	Frame	1	71.0	80.9	77.3	50.9	61.6	57.7	57.5	74.2	58.6	83.8	88.4	83.1
+MPT (Ours)	Frame	1	70.7	80.5	76.9	52.8	64.0	60.7	57.8	74.8	59.5	84.6	89.1	83.8
SeqTrack-L384 (Chen et al., 2023)	Frame	2	72.5	81.4	79.2	50.4	61.2	57.2	<u>59.6</u>	<u>76.6</u>	<u>63.5</u>	85.6	90.0	85.9
+MPT (Ours)	Frame	2	73.9	83.2	81.0	<u>51.7</u>	<u>62.7</u>	<u>58.8</u>	60.4	77.5	64.2	<u>85.8</u>	<u>90.1</u>	86.2
ARTrack-L384 (Wei et al., 2023)	Seq	1	73.1	82.0	80.2	52.0	62.4	59.0	60.8	77.5	64.5	85.4	90.2	85.8
ODTrack-B384 (Zheng et al., 2024)	Seq	3	73.2	83.2	80.6	52.4	63.9	60.1	60.9	-	-	85.1	90.1	84.9
ARTrackV2-L384 (Bai et al., 2024)	Seq	2	73.6	82.8	81.1	53.4	63.7	60.2	61.6	-	-	86.1	90.4	86.2
ODTrack-L384 (Zheng et al., 2024)	Seq	3	74.0	84.2	82.3	53.9	65.4	61.7	61.7	-	-	86.1	91.0	86.7
Average gain	-	-	+0.4%	6 +0.8% ·	+0.6%	+1.6%	6 +2.1%	+2.4%	+0.6%	6 +1.0%	+1.0%	+0.4%	+0.4%	+0.3%

Table 2. **Comparisons on four tracking benchmarks.** Top two results (except for sequential-training methods) are marked in **bold** and <u>underlined</u>. For a fair comparison, we list the training strategy (Train) and template amounts (Temp.) of each tracker.

Table 3. Efficiency of training and inference. Two A100 GPUs are used with 128 batch sizes (BS), except for ARTrack with 8 BS. Time_{tr} and N_T denote training time and frame numbers of temporal information.

Tracker (BS)	FLOPs	Params	FPS	Mem	Time_{tr}	N_T
OSTrack (128)	31.2G	92M	94	35G	38h	1
+ MPT (128)	35.9G	105M	86	6G	5h	30
ARTrack (8)	55.2G	202M	26	50G	26+53h	7
+ MPT (128)	56.8G	215M	26	21G	12h	30

6G memory. It demonstrates that our MPT is training efficient, and offers incremental benefits to various trackers with minimal training resources.

4.3. Ablation Study and Analysis

To deeply analyze our MPT, we conduct comprehensive ablation studies on $LaSOT_{EXT}$ and VOT2022, choosing OSTrack-B256+MPT as our baseline (denoted as Ours). We present AUC and EAO for two benchmarks, respectively, and also the averaged performance change (denoted as Δ).

Positional encoding. To explore impacts of the adopted

Table 4. **Impact of positional encodings,** including respective role of each PE and comparisons of TPE variants.

		1					
	1	2	3	4	5	6	$\overline{\mathcal{O}}$
	Ours	w/o SPE	w/o TPE	w/o P	Init _{ran}	Init _{lin}	Non-Learn
LaSOT _{EXT}	48.7	44.8	47.5	48.1	45.9	47.4	47.4
VOT2022	0.572	0.556	0.547	0.573	0.567	0.561	0.552
Δ	-	-2.8%	-1.9%	-0.3%	-1.7%	-1.2%	-1.7%

three positional encodings, we individually remove one type of PE, shown as Tab. 4. In our motion encoder, the spatial PE (SPE) is mainly used to encode each point coordinate to the visual embedding space, facilitating the alignment of motion and visual features in terms of spatial positions. When we replace our SPE with a simple MLP (⁽²⁾), the tracking performance decreases by 2.8% on average. Besides, our temporal PE (TPE) integrates temporal orders to motion tokens in a non-linear way, and the employed point embeddings (P) identify the type of corner points for motion tokens. Removing any of them (⁽³⁾) and ⁽⁴⁾) results in a certain degree of performance decline. These results indicate that each type of PE plays a crucial role in the performance of

	-				-	
	Ours	$\mathcal{L}_{\mathrm{Tr}} + \mathcal{L}_{\mathrm{M}}$	$\mathcal{L}_{\mathrm{Tr}}$	Motion	Vision	w/o Weight
$\begin{array}{c} \text{LaSOT}_{\text{EXT}} \\ \text{VOT2022} \\ \Delta \end{array}$	48.7 0.572 -	48.6 0.558 -0.8%	45.1 0.556 -2.6%	48.5 0.525 -2.5%	46.9 0.530 -3.0%	46.4 0.568 -1.4%
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	cight		N N N N N N N N N N N N N N N N N N N		WW.	M
0 2 0.8 - 0.6 - 0.2 - 0.0 -	**************************************		Frame ⁸⁰⁰	And And	JIZOO	I ADO

Table 5. Impact of different losses and components.

Figure 3. Illustration of adaptive weights, and tracking performance based on different cues.

our MPT, particularly SPE and TPE.

Furthermore, we adopt a non-linear initialization on a learnable embedding for our TPE (formulated as Eq. 3) to make the model dynamically focus on the significant motion information. To explore the impact of different TPE initializations, we compare random (⑤) and linear (⑥) initializations with our non-linear method(①). It shows that our non-linear TPE initialization performs superior to others. Besides, to investigate the efficacy of the learnable attribute in our TPE, we also compare our method with a variant with nonlearnable TPE (⑦). Disabling the learning capability of TPE results in a performance degradation of 1.7%.

Loss function. We formulate three loss terms for model optimization. In addition to the baseline tracking loss \mathcal{L}_{Tr} , we introduce the other two loss terms to enhance supervision. Among them, the motion regression loss \mathcal{L}_M guides Rep. tokens to extract essential motion patterns from the trajectory. The adaptive weight loss \mathcal{L}_W is employed to explicitly supervise the weight prediction, allowing the model to adaptively adjust its reliance on visual or motion cues. Results reported in Tab. 5 demonstrate the significance of both \mathcal{L}_M and \mathcal{L}_W . The combined utilization of all three losses (Ours) boosts the model to achieve superior performance.

Adaptive weight mechanism. Results in Tab. 5 (right) demonstrate the respective roles of visual and motion inputs, as well as the effectiveness of the proposed adaptive weight mechanism. "Motion" represents the tracking results directly predicted by the motion head $Head_M$, while "Vision" shows the results of the baseline vision-based tracker. We also implement a variant of MPT without the adaptive weight mechanism, denoted as "w/o Weight". That is, instead of calculating the weighted average with the original visual features, we directly use the output of the fusion



Figure 4. Success rate of among varying trajectory qualities.



Figure 5. IoU distributions of average trajectory and the last frame in success and failure cases.

decoder for tracking predictions. As observed, our MPT, compared to the tracker relying solely on visual information, yields more robust tracking results by incorporating motion prompts. Additionally, the adaptive weight mechanism contributes to our method achieving further improvements.

Besides, we visualize frame-level qualities of tracking results predicted by the aforementioned three variant models, namely "Vision", "Motion", and our MPT. As shown in Fig. 3, when encountering plenty of distractors, the prediction of the vision model (blue curve) is notably unstable. In contrast, our MPT (red curve) can exhibit superior and robust performance. Furthermore, the values of our dynamic weights (green curve) are consistent with the quality of motion predictions (orange curve). This demonstrates that our adaptive weight mechanism can achieve a dynamic balance between visual and motion cues.

Sensitivity to the trajectory quality. To further analyze the sensitivity of our MPT to varying noisy trajectories, we evaluate the success rate (IoU > 0.5) under different quality trajectory inputs on LaSOT_{EXT}, as shown in Fig. 4. Here, [a,b) represents the range where the average IoU of the trajectory is no less than a and less than b. We can find that the success rate increases significantly with the improvement of trajectory quality, especially when the trajectory IoU is in the range of [0.6, 1.0). Besides, our method shows a certain level of robustness to noisy trajectory inputs, obtaining a

	υ		2	
	OSTrack	Ours	FinetuneAll	FinetuneHead
LaSOT _{EXT}	46.9	48.7	48.3	48.1
VOT2022	0.530	0.572	0.567	0.548
Mem (G)	35.0	6.1	37.5	7.7
$\operatorname{Time}_{tr}(h)$	38	5	8	6

Table 6. **Impact of fine-tuning strategies,** in terms of tracking performance and training efficiency.

success rate of 21% even when the trajectory IoU is in the range of [0.1, 0.2).

In addition, to deeply explore the success and failure cases of our MPT, we separately evaluate the distribution of trajectory IoU (IoUtrai) and tracking IoU of the last frame (IoU_{last}) , in both conditions on $LaSOT_{EXT}$. Here, we define a success case when the predicted IoU of our MPT exceeds that of the visual baseline by more than 0.3. Conversely, when the predicted IoU of our MPT is lower than that of the visual baseline by more than 0.3, we define it as a failure case. As shown in Fig. 5, in success cases, the input trajectory IoU primarily falls within a relatively broader range of [0.6, 1.0), especially in [0.8, 0.9). In failure cases, the input trajectory IoU mainly falls within the range of [0.0, 0.2), particularly in [0.0, 0.1). This indicates that addressing misleading tracking issues caused by extremely poor-quality trajectories remains a challenge for our method. This is a limitation of our method that requires future research.

Fine-tuning strategies. Our method freezes all of the baseline parameters, and only fine-tunes parameters of MPT during training. We also evaluate two variants, shown in Tab. 6. FinetuneAll fine-tunes both MPT and baseline parameters, while FinetuneHead fine-tunes parameters of MPT and the tracking head of the baseline, *i.e.*, only freezing the backbone of the baseline model. Our method slightly outperforms the other two variants, and is more efficient in terms of memory footprint and training time. The potential reason is that freezing the baseline parameters enables the model to focus on learning motion cues and the fusion mechanism.

Visualization of cross-attention maps. For a more straightforward understanding of how our MPT fuses visual and motion features, we visualize the cross-attention maps (averaged over heads) of the last decoder block, shown as Fig. 6. Among them, "Attn_{t1}" and "Attn_{br}" represent cross-attention maps between visual features and two representative motion tokens (RT_{t1} and RT_{t1}), respectively. In addition, we also visualize the score maps predicted by the vision-based baseline tracker (ScoreMap_V), and our MPT model (ScoreMap_{MPT}). We find that it is difficult to distinguish distractors based on a solely visual model. Due to the complement of motion prompts, our MPT pays more attention to the top-left and bottom-right corner points of the interested target. This enables the model to weaken the



Figure 6. Visualization of cross-attention maps, and discriminative comparison of score maps.

response to distractors, thereby achieving robust tracking.

Due to space limitations, more visualizations can be found in Appendix E, including the optimizing process during training (Fig. 11), qualitative comparisons (Fig. 13), and some failure cases (Fig. 12).

Limitations. Although our MPT excels in training efficiency and tracking robustness, its tracking accuracy is slightly inferior to those sequential training methods. We will further explore effective methods to capture dynamic relations in efficient prompting. In addition, as shown in Fig. 5, addressing misleading tracking issues caused by extremely poor-quality trajectories remains a challenge for our method. which requires future research.

5. Conclusion

In this paper, we present a flexible and efficient motion prompt tracking method. It can complement existing visionbased trackers by learning motion prompts, thereby maintaining robust tracking in challenging scenarios. Our method includes a motion encoder with three different positional encodings, a Transformer-based fusion decoder, and an adaptive weight mechanism. Motion trajectories of the object are independently encoded and then adaptively fused with visual features. We integrate our method into five visual tracking models. Experiments on seven datasets show that our method efficiently enhances tracking performance within minimal computational costs and memory usage. We hope that our new insights into temporal tracking and prompt tracking will leverage future research on these topics.

Acknowledgments

The computational resources were provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at C3SE partially funded by the Swedish Research Council grant 2022-06725. This work was supported in part by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP), funded by Knut and Alice Wallenberg Foundation, in part by the strategic research environment ELLIIT funded by the Swedish government, and the Swedish Research Council grant 202204266, in part by National Natural Science Foundations of China (No. 62402084 and No. 62176041), in part by Liaoning Province Science and Technology Plan (No. 2024JH2/102600040), and in part by China Postdoctoral Science Foundation (No. 2024M750319).

Impact Statement

This work advances the field of visual object tracking through efficient motion prompt learning. The technique may benefit real-time applications such as surveillance systems.

References

- Bai, Y., Zhao, Z., Gong, Y., and Wei, X. ARTrackV2: Prompting autoregressive tracker where to look and how to describe. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 19048–19057, 2024.
- Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, pp. 850–865, 2016.
- Bhat, G., Danelljan, M., Gool, L. V., and Timofte, R. Learning discriminative model prediction for tracking. In *IEEE International Conference on Computer Vision*, pp. 6182– 6191, 2019.
- Cai, Y., Liu, J., Tang, J., and Wu, G. Robust object modeling for visual tracking. In *IEEE International Conference on Computer Vision*, pp. 9589–9600, 2023.
- Cao, Z., Huang, Z., Pan, L., Zhang, S., Liu, Z., and Fu, C. Towards real-world visual tracking with temporal contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- Chen, B., Li, P., Bai, L., Qiao, L., Shen, Q., Li, B., Gan, W., Wu, W., and Ouyang, W. Backbone is all your need: A simplified architecture for visual object tracking. In *European Conference on Computer Vision*, pp. 375–392, 2022a.
- Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., and Lu, H. Transformer tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8126–8135, 2021.
- Chen, X., Peng, H., Wang, D., Lu, H., and Hu, H. SeqTrack: Sequence to sequence learning for visual object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 14572–14581, 2023.
- Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., and Qiao, Y. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022b.

- Cui, Y., Jiang, C., Wang, L., and Wu, G. MixFormer: Endto-end tracking with iterative mixed attention. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13608–13618, 2022.
- Cui, Y., Jiang, C., Wu, G., and Wang, L. MixFormer: End-toend tracking with iterative mixed attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46 (6):4129–4146, 2024.
- Dai, K., Wang, D., Lu, H., Sun, C., and Li, J. Visual tracking via adaptive spatially-regularized correlation filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4670–4679, 2019.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representation*, 2021.
- Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., and Ling, H. LaSOT: A high-quality benchmark for large-scale single object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5374–5383, 2019.
- Fan, H., Bai, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Huang, M., Liu, J., Xu, Y., et al. LaSOT: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129:439–461, 2021.
- Gao, S., Zhou, C., Ma, C., Wang, X., and Yuan, J. AiATrack: Attention in attention for transformer visual tracking. In *European Conference on Computer Vision*, pp. 146–164, 2022.
- Huang, L., Zhao, X., and Huang, K. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1562–1577, 2019.
- Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727, 2022.
- Ju, C., Han, T., Zheng, K., Zhang, Y., and Xie, W. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision*, pp. 105– 124. Springer, 2022.
- Kim, M., Lee, S., Ok, J., Han, B., and Cho, M. Towards sequence-level training for visual tracking. In *European Conference on Computer Vision*, pp. 534–551, 2022.

- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. In *IEEE International Conference on Computer Vision*, pp. 4015–4026, 2023.
- Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., [°]Cehovin Zajc, L., Vojir, T., Bhat, G., Lukezic, A., Eldesokey, A., et al. The sixth visual object tracking VOT2018 challenge results. In *European Conference on Computer Vision*, 2018.
- Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.-K., Danelljan, M., Zajc, L. Č., Lukežič, A., Drbohlav, O., et al. The eighth visual object tracking VOT2020 challenge results. In *European Conference on Computer Vision*, pp. 547–601, 2020.
- 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. The tenth visual object tracking vot2022 challenge results. In *European Conference on Computer Vision*, pp. 431–460, 2022.
- Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., and Yan, J. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4282–4291, 2019.
- Lin, L., Fan, H., Xu, Y., and Ling, H. SwinTrack: A simple and strong baseline for transformer tracking. In Advances in Neural Information Processing Systems, 2022.
- Liu, C., Yuan, Y., Chen, X., Lu, H., and Wang, D. Spatialtemporal initialization dilemma: towards realistic visual tracking. *Visual Intelligence*, 2(1):35, 2024.
- Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In Advances in Neural Information Processing Systems, volume 36, pp. 34892–34916, 2023a.
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys, 55(9):1–35, 2023b.
- Mayer, C., Danelljan, M., Bhat, G., Paul, M., Paudel, D. P., Yu, F., and Van Gool, L. Transforming model prediction for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8731–8740, 2022.
- Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., and Ghanem, B. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In *European Conference on Computer Vision*, pp. 300–317, 2018.
- Nyquist, H. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

- Ramachandran, V. The neurobiology of perception. *Perception*, 14(2):97–103, 1985.
- Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., and Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 658–666, 2019.
- Sokhandan, A. and Monadjemi, A. Visual tracking in video sequences based on biologically inspired mechanisms. *Computer Vision and Image Understanding*, 239:102724, 2024.
- Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. In Advances in Neural Information Processing Systems, volume 33, pp. 7537–7547, 2020.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.
- Voigtlaender, P., Luiten, J., Torr, P. H., and Leibe, B. Siam R-CNN: Visual tracking by re-detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6578–6588, 2020.
- Wang, X., Shu, X., Zhang, Z., Jiang, B., Wang, Y., Tian, Y., and Wu, F. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13763–13773, 2021.
- Wei, X., Bai, Y., Zheng, Y., Shi, D., and Gong, Y. Autoregressive visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9697–9706, 2023.
- Weng, S.-K., Kuo, C.-M., and Tu, S.-K. Video object tracking using adaptive kalman filter. *Journal of Visual Communication and Image Representation*, 17(6):1190–1208, 2006.
- Yan, B., Peng, H., Fu, J., Wang, D., and Lu, H. Learning spatio-temporal transformer for visual tracking. In *IEEE International Conference on Computer Vision*, pp. 10448– 10457, 2021.
- Yang, J., Li, Z., Zheng, F., Leonardis, A., and Song, J. Prompting for multi-modal tracking. In ACM International Conference on Multimedia, pp. 3492–3500, 2022.
- Ye, B., Chang, H., Ma, B., Shan, S., and Chen, X. Joint feature learning and relation modeling for tracking: A

one-stream framework. In *European Conference on Computer Vision*, pp. 341–357, 2022.

- Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. CutMix: Regularization strategy to train strong classifiers with localizable features. In *IEEE International Conference on Computer Vision*, pp. 6023–6032, 2019.
- Zhang, M., Zhang, Q., Song, W., Huang, D., and He, Q. Promptvt: Prompting for efficient and accurate visual tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.
- Zhang, S., Chan, S.-C., Liao, B., and Tsui, K. M. A new visual object tracking algorithm using bayesian kalman filter. In *IEEE International Symposium on Circuits and Systems*, pp. 522–525, 2014.
- Zhao, J., Dai, K., Zhang, P., Wang, D., and Lu, H. Robust online tracking with meta-updater. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6168– 6182, 2022a.
- Zhao, J., Zhang, J., Li, D., and Wang, D. Vision-based anti-uav detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):25323–25334, 2022b.
- Zheng, Y., Zhong, B., Liang, Q., Mo, Z., Zhang, S., and Li, X. ODTrack: Online dense temporal token learning for visual tracking. In AAAI Conference on Artificial Intelligence, 2024.
- Zhu, J., Lai, S., Chen, X., Wang, D., and Lu, H. Visual prompt multi-modal tracking. In *IEEE Conference* on Computer Vision and Pattern Recognition, pp. 9516– 9526, 2023.

Appendix

This appendix contains proof of non-linear temporal positional encoding in Section A, additional implementation details in Section B, extended ablation studies in Section C, in-depth performance analysis in Section D, and more visualizations in Section E.



A. Proof of Non-Linear Temporal Positional Encoding

Figure 7. Impacts of different parameter α on VOT2022.

To allow our model to perceive temporal orders of motion tokens, we adopt a learnable temporal positional encoding and initialize it in the following non-linear manner:

$$PE(t,2i) = \sin\left(\frac{\alpha \lg(t+1)}{n^{2i/d}}\right),$$

$$PE(t,2i+1) = \cos\left(\frac{\alpha \lg(t+1)}{n^{2i/d}}\right).$$
(7)

The parameter α is employed to control the frequency range, and set to 7.23 based on the Nyquist frequency (Nyquist, 1928). The proof of this setting is presented as follows.

According to the theory of Nyquist frequency, a signal should be sampled at a rate at least twice its highest frequency to avoid aliasing. Due to the logarithmic mapping of the time parameter t, the frequency ω of Eq. 7 changes in a non-linear way and needs to be determined using the derivative (note that $\lg(t+1) = \ln(t+1)/\ln(10)$):

$$\omega(t,i) = \frac{\partial_t PE(t,2i)}{PE(t,2i+1)} = \frac{\alpha}{\ln(10)(t+1)n^{2i/d}}.$$
(8)

Its maximum is attained at (t, i) = (0, 0). From $\omega(0, 0) = \pi$, we obtain $\alpha = \pi \ln(10) \approx 7.23$.

Moreover, we also verify its experimental effectiveness by comparing it with other parameter values. As shown in Fig. 7, results on VOT2022 (Kristan et al., 2022) show that the best performance is attained at $\alpha = 7.23$. Other parameter values cause slight performance decays, but they are still superior to the baseline tracker (marked as the horizontal dashed line).

B. Implementation Details

Model. For our motion encoder, we encode each 2-dimensional coordinate into a motion token with the same dimension as the baseline visual features, which are 768 and 256 for OSTrack-B / ARTrack-B, and SeqTrack-B, respectively. The lightweight fusion decoder is implemented as a two-layer Transformer network. For each layer, the number of attention heads is 8, and the hidden size of MLP is set to 1024 and 256 for OSTrack / ARTrack and SeqTrack, respectively. The weight head $Head_W$ and motion head $Head_M$ are both implemented by a two-layer MLP, where the hidden size is 256.

Training. The model is trained for 60 epochs with 60k image pairs per epoch. We set the batch size to 128, and the learning rate is decreased by a factor of 10 after 40 epochs. The initial learning rate and other training settings are set the same as



Figure 8. Impacts of data augmentations on VOT2022.

corresponding baseline trackers. For the motion input, we sparsely sample a T-frames trajectory either before or after the current search frame. In a trajectory, we employ equally spaced sparse sampling to preserve the original motion pattern of the target. Such an offline trajectory generation strategy not only simplifies and accelerates the training process, but also allows the model to confront real tracking noises and learn to adapt to motion trajectories with varying qualities. In addition, we adopt CutMix strategy (Yun et al., 2019) on search images to simulate challenges like background clutter. Specifically, we crop the object area from another frame, and paste it on a random position of the search image. This data augmentation offers more chances for our model to learn how to effectively fuse and utilize extra motion information. The following Section will analyze the impacts of the adopted data augmentations.



Figure 9. Impact of the trajectory length. Our MPT obtains the best performance when the length of trajectories T = 30.

C. Extended Ablation Study

C.1. Analysis of Data Augmentations

Fig. 8 illustrates the individual impacts of CutMix and sparse sampling. For the CutMix, training our method without using CutMix strategy (*i.e.* probability is set to 0) obtains slightly inferior performance than the baseline visual tracker. The potential reason is that most original training samples are easy for the baseline visual model, limiting chances for our motion model to learn the ability to complement and optimize tracking predictions. CutMix, however, can simulate hard scenes by introducing distractors and slight occlusion, which mitigates this training issue. Fig. 8(a) shows that the best performance of our model is attained when the probability of CutMix is set to 0.5. Therefore, we set the probability of 0.5 as the default in our experiments.

Efficient Motion	Prompt	Learning	for Re	bust `	Visual	Tracking
		···				

		LaSOT		LaSOT _{EXT}		TNL2K		ALL		
		Num	Gain	Num	Gain	Num	Gain	Num	Gain	
OSTrack D256	1	50	+4.8	80	+4.8	284	+2.3	414	+4.0	
OS Irack-B250	2	230	-0.7	70	-1.6	416	-1.4	716	-1.2	
OSTrack-B384	1	40	+5.6	71	+4.5	252	+2.1	363	+4.1	
	2	240	-1.3	79	-0.5	448	-0.7	767	-0.8	
	1	44	+5.5	70	+2.8	271	+3.8	385	+4.0	
SeqTrack-B230	2	236	-0.2	80	-0.4	429	-0.4	745	-0.3	
SagTraalt I 294	1	38	+9.3	76	+4.4	234	+2.8	348	+5.5	
SeqTrack-L384	2	242	+0.2	74	-1.8	466	-0.3	782	-0.6	
ADTreals D256	1	43	+5.9	78	+4.8	255	+3.2	376	+4.6	
ARTrack-B256	2	237	-1.9	72	-0.9	445	-0.8	754	-1.2	

Table 7. Quantitative analysis of performance bias

For sparse sampling, increasing the range of sparseness for trajectory sampling enables the model to learn from more diverse motion inputs, but it concurrently means that our model faces more intricate and challenging motion patterns during the training. As shown in Fig. 8(b), the sparseness of 5 is an optimized choice, which is also our default setting.

C.2. Impact of Trajectory Length

Fig. 9 illustrates the impact of the motion trajectory length T on the performance of our MPT. Our method exhibits improved performance as T increases, and reaches its peak at T = 30. Since the longer trajectory might bring more motion noises, further increasing the trajectory length results in slight performance degradation. Therefore, we set T = 30 in our experiments. This ablation study confirms our conjecture that long-term temporal information can provide richer tracking cues. Different from prior works (Kim et al., 2022; Wei et al., 2023) employing heavy temporal processing mechanisms, our lightweight MPT independently encodes motion inputs using negligible computing resources, which allows the model to manage long-term motion trajectories.

D. In-depth Performance Analysis

D.1. In-depth Analysis about Performance Bias

Aiming to reduce tracking failures by motion cues, our MPT can significantly improve tracking robustness over accuracy. Our advantages are more evident in hard scenarios (like VOTs, $LaSOT_{EXT}$). This may cause a slight bias in the overall performance across different trackers or benchmarks. To give evidence for this conjecture, we dynamically split each dataset into two subsets based on performance of baselines, namely hard / easy sets. As shown in Tab. 7, the hard set (①) includes hard videos with baseline AUCs below 50.0%, while the easy set (②) includes the remaining videos. Hard set proportions of LaSOT, TNL2K, LaSOT_{EXT} are around 15%, 30%, 50%. This aligns with the trend in our average performance gains: 0.4%, 0.6%, 1.6%. Moreover, our MPT shows better consistency in hard / easy sets, achieving average AUC changes of +4.4% / -0.8%, with small standard deviations of 0.6 / 0.4 across five baselines.

D.2. Attribute Analysis

For a more intuitive observation of the impacts of our MPT on different attributes, we report performance gains achieved by our MPT across different attributes on $LaSOT_{EXT}$ (Fan et al., 2021), shown as Fig. 10. As we can see, our MPT significantly enhances the baseline tracker when faced with the challenge of background clutter, obtaining 6.3% AUC improvements. Besides, challenges like viewpoint change, partial occlusion, low resolution, deformation, and full occlusion can also be better addressed by our MPT.



Figure 10. Performance gains of our method on each attribute.

E. Visualizations

E.1. Visualization of the Training Process

As shown in Fig. 11, we present the optimizing process of our model by visualizing dynamic changes of (a) loss values, (b) Intersection-over-Union (IoU), and (c) location accuracy of the score map (Acc). Here, we consider the location prediction to be accurate if the position of the maximum value in the score map coincides with the center of the target groundtruth. As we can see, due to fine-tuning few parameters, the model exhibits fast convergence during the training process. Moreover, compared with the baseline visual model OSTrack-B256 (Ye et al., 2022), our motion prompt-based model yields a more accurate score map (see Fig. 11(c)), consequently leading to a more precise prediction of the object position (see Fig. 11(b)). These curves further substantiate the efficacy of our MPT method in terms of both training efficiency and performance.



Figure 11. Visualization of optimizing process of our model during training.

E.2. Qualitative Comparison

To intuitively demonstrate the effectiveness of our MPT, we make a qualitative comparison of our methods with corresponding baseline trackers and other state-of-the-art (SOTA) trackers. As shown in Fig. 13, our MPT allows SOTA vision-based baseline trackers to perform more robustly when faced with challenges, like distractors, occlusion, severe appearance changes, fast motion, etc.



Figure 12. Several failure cases of our method ("_MPT").

Moreover, we make a video (please find it at https://github.com/zj5559/Motion-Prompt-Tracking) to show more qualitative comparisons on three challenging sequences. For each sequence, we visualize bounding boxes predicted by our method ("+MPT") and the corresponding baseline tracker OSTrack. We also compare score maps output by these two trackers. The historical trajectories are plotted in our score maps ("Feat_MPT"). Compared with the baseline tracker, our MPT can utilize object motion cues to help the tracker recognize the interested object from multiple distractors. Besides, when faced with severe occlusion, our MPT can predict the next position of the object based mainly on the motion prompt, thereby avoiding tracking drift.

However, since motion trajectories often include inherent noises like wrong observations, low-quality motion cues sometimes interfere with visual cues. We propose an adaptive weight mechanism to mitigate this phenomenon, but some tracking failures still inevitably happen. Some failure cases of our method are shown in Fig. 12.



Figure 13. Qualitative comparisons of our trackers ("_MPT") with other SOTA trackers.