TGTP: GENERAL TRAJECTORY-GUIDED TOKEN PRUNING FOR VISUAL TRACKING VIA REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Token pruning in visual tracking aims to reduce computational cost by removing redundant tokens, with the key challenge lying in reliable token importance estimation. Existing methods often rely on current-frame cues such as attention responses, token similarity, or static pruning criteria, while making limited use of the temporal continuity inherent to tracking. Since target position, scale, and motion evolve continuously over time, historical trajectories can provide useful priors for token selection. To this end, we propose Trajectory-Guided Token Pruning (TGTP), a general reinforcement learning-based token pruning framework for visual tracking. The core of TGTP is a Trajectory-Guided Reinforcement Learning (TGRL) module, which formulates token pruning as a sequential decision-making problem. TGRL constructs states from recent target trajectories and learns pruning actions through a Trajectory Policy Network, predicting motion priors, motion uncertainty, and token keep ratios for the current frame. The policy is optimized with rewards that balance tracking accuracy, FLOPs, uncertainty regularization, and target-token preservation. As a lightweight plug-in module, TGTP can be integrated into different tracking architectures without redesigning their original structures. Experiments on multiple single-modal and multi-modal baselines show that TGTP substantially reduces search tokens and model FLOPs while maintaining or improving tracking performance, demonstrating its effectiveness and generality for computation-efficient visual tracking.

1 INTRODUCTION

Visual tracking aims to localize a specified target throughout a video sequence. Recent token-based trackers Kang et al. (2025a); Zeng et al. (2025); Wu et al. (2026b); Yang et al. (2026a) have achieved strong performance by modeling interactions between template and search regions. However, these methods usually retain a large number of search tokens during inference, resulting in considerable computational cost. In practice, the target often occupies only a small portion of the search region, especially in UAV tracking Xue et al. (2025); Yang et al. (2026b) and long-range surveillance Wu et al. (2025) scenarios. As a result, many background tokens contribute little to target localization but are still involved in network computation. This increases model FLOPs and may introduce redundant background information. Therefore, reducing redundant tokens while preserving tracking performance is an important problem for computation-efficient visual tracking.

Token pruning provides a direct way to reduce the computational cost of visual tracking. Its key challenge is to estimate token importance reliably, such that target-related tokens are retained and redundant background tokens are removed. Existing pruning methods Bolya et al. (2022); Liang et al. (2022); Ye et al. (2022); Wu et al. (2026a) typically determine token importance using current-frame cues, including attention responses, token similarity, or predefined pruning rules. While these cues can highlight target-related regions, they mainly reflect single-frame evidence and make limited use of the temporal continuity inherent to tracking. Since target position, scale, and motion evolve continuously across adjacent frames, relying only on current-frame cues may limit the adaptability of pruning policies under motion variation, scale change, occlusion, and background interference.

Based on this observation, we argue that historical trajectories provide useful priors for token importance estimation. The target center, scale variation, and motion pattern in recent frames can indicate

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

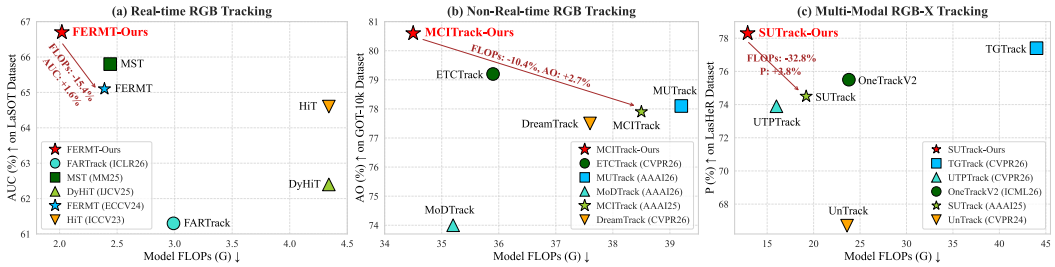


Figure 1: Performance-FLOPs comparison across real-time RGB, non-real-time RGB, and multi-modal RGB-X tracking. TGTP consistently reduces model FLOPs while maintaining or improving tracking performance when integrated into representative visual trackers.

where the target is likely to appear in the current frame, thereby helping identify search tokens that are more likely to be target-related. Compared with single-frame attention or static pruning rules, trajectory information naturally captures cross-frame dynamics and better matches the sequential nature of visual tracking. Incorporating such trajectory priors into token pruning can therefore improve the adaptability of pruning policies to target dynamics and reduce unnecessary computation.

To this end, we propose Trajectory-Guided Token Pruning (TGTP), a general trajectory-guided token pruning framework for visual tracking. TGTP formulates token pruning as a reinforcement learning driven sequential decision-making problem, whose core is a Trajectory-Guided Reinforcement Learning (TGRL) module. TGRL first constructs a state representation from the historical target trajectories in the recent K frames, including target location, scale, and their temporal variations. Then, a Trajectory Policy Network samples actions from the state to predict the motion prior, motion uncertainty, and token keep ratio for the current frame. The policy is optimized by a reward function that jointly considers tracking accuracy, FLOPs, motion uncertainty, and target-token preservation.

TGTP is designed as a general plug-in module. It is not tied to a specific tracker architecture and can be integrated into existing tracking frameworks with minimal structural modification. By using trajectory priors to assist token importance estimation, TGTP performs pruning at selected network layers and is applicable to both single-modal and multi-modal trackers. As shown in Fig. 1, integrating TGTP into representative real-time RGB, non-real-time RGB, and multi-modal RGB-X trackers consistently reduces model FLOPs while maintaining or improving tracking performance. Comprehensive experiments further validate the effectiveness and generality of TGTP in reducing redundant tokens and computational cost for visual tracking.

The main contributions of this work are summarized as follows:

- We propose Trajectory-Guided Token Pruning (TGTP), a general token pruning framework for visual tracking, which reduces computational cost while maintaining tracking accuracy.
- We design a Trajectory-Guided Reinforcement Learning (TGRL) module, which uses historical target trajectories to construct states and learns motion priors, motion uncertainty, and token keep ratios through reinforcement learning.
- We integrate TGTP into multiple single-modal and multi-modal tracking baselines, demonstrating its effectiveness and generality in reducing search tokens and model FLOPs.

2 Related Work

2.1 Visual Tracking

Visual tracking aims to localize a target object in each frame of a video sequence. Existing trackers can be broadly divided into single-modal and multi-modal trackers. Single-modal RGB trackers focus on tracking from visible images and have been developed along both real-time and non-real-time directions. Real-time trackers Zheng et al. (2024); Kang et al. (2025b); Zhu et al. (2025); Wang et al. (2025) emphasize efficient inference, while non-real-time trackers Xie et al. (2025); Cai et al. (2025); Huang et al. (2026) usually pursue stronger representation and more accurate localization. Beyond RGB tracking, multi-modal trackers Su et al. (2026); Kang et al. (2026); Geng et al. (2026); Hong et al. (2026) exploit complementary information from additional modalities, such as thermal infrared, depth, and event streams, leading to RGB-T, RGB-D, and RGB-E tracking.

Despite their progress, many recent trackers still process a large number of search-region tokens to maintain tracking accuracy. This is particularly costly for token-based methods, where background tokens are repeatedly involved in feature interaction and network propagation. Since the target usually occupies only a small portion of the search region, many tokens contribute little to final localization but increase FLOPs. This issue becomes more evident in high-resolution tracking and multi-modal tracking, where additional modalities further increase computational burden. Different from designing a tracker-specific lightweight architecture, our work focuses on a general token pruning module that can be plugged into existing trackers. By exploiting historical target trajectories, the proposed TGTP reduces redundant search tokens while preserving target-related information.

2.2 Visual Token Pruning

Token pruning Bolya et al. (2022); Liang et al. (2022) has been widely studied as an effective strategy for improving the efficiency of vision models. In token-based architectures, redundant tokens can be removed, merged, or reorganized to reduce computation in subsequent layers. Existing methods often estimate token importance using attention responses, feature similarity, prediction confidence, or lightweight scoring networks. In visual tracking, token pruning is especially attractive because the search region usually contains substantial background redundancy. Recent tracking-oriented pruning methods Ye et al. (2022); Wu et al. (2026c;a) attempt to remove less informative search tokens during feature extraction, thereby reducing FLOPs while maintaining localization accuracy.

However, most existing pruning strategies make decisions mainly from current-frame cues, such as attention maps, token similarity, or fixed pruning ratios. Although these cues are useful, they provide limited modeling of the temporal continuity that is inherent to visual tracking. The target position, scale, and motion evolve across frames, and such historical trajectories can offer strong priors for identifying target-related tokens in the current search region. Moreover, fixed or hand-designed pruning rules may be insufficient to adapt to different target dynamics and scene difficulties. In contrast, TGTP formulates token pruning as a reinforcement learning-based sequential decision-making problem. Through the TGRL module, it constructs states from historical trajectories and learns actions that predict motion priors, motion uncertainty, and token keep ratios, enabling trajectory-aware token pruning across different tracking architectures.

3 The Proposed Method

3.1 Problem Formulation

Given a template image z and the search region x_t in the t-th frame, visual tracking aims to predict the target bounding box $\hat{\mathbf{b}}_t$ in the current frame. After patch embedding, the template image and search region are represented as template tokens $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{N_z}$ and search tokens $\mathbf{X}_t = \{\mathbf{x}_{t,i}\}_{i=1}^{N_x}$, where N_z and N_x denote the numbers of template and search tokens, respectively. Existing token-based trackers usually perform feature interaction over all search tokens in the backbone or encoder, and then predict the target box through a tracking head. However, in visual tracking, the target often occupies only a small portion of the search region. Many background tokens contribute little to target localization but are still involved in subsequent computation, leading to redundant FLOPs.

Therefore, the key problem of token pruning in visual tracking is how to select a target-related retained set from search tokens, such that the tracker can reduce search tokens and computational cost while preserving accurate target localization. Let L_p denote the set of selected pruning layers and $N_{t,\ell}^x$ denote the number of search tokens before pruning at layer $\ell \in L_p$. Let $\widetilde{\mathbf{A}}_{t,i}^{\ell}$ denote the importance score of the i-th search token at layer ℓ. The token pruning process can be expressed as:

$$\mathcal{M}_t^{\ell} = \mathrm{TopK}\left(\{\widetilde{\mathbf{A}}_{t,i}^{\ell}\}_{i=1}^{N_{t,\ell}^x}, N_{t,\ell}^{\mathrm{keep}}\right), \qquad N_{t,\ell}^{\mathrm{keep}} = \left\lceil \gamma_t N_{t,\ell}^x \right\rceil,$$

where $\mathcal{M}_t^{\ell}$ is the retained search-token set at layer ℓ and γ_t denotes the token keep ratio in the current frame. For clarity, the layer index ℓ is omitted when there is no ambiguity.

Different from existing methods that mainly rely on current-frame attention responses or static pruning rules, we argue that token pruning in visual tracking should exploit the temporal evolution of the target across consecutive frames. Since the target position, scale, and motion trend evolve from

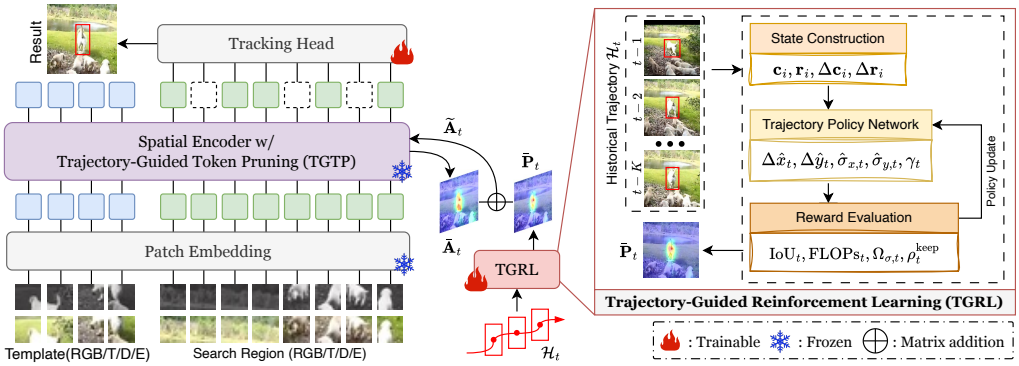


Figure 2: Overview of integrating TGTP into existing visual trackers. TGTP uses a trajectory-guided reinforcement learning module to generate pruning actions from historical target trajectories, and combines trajectory priors with template-search attention to estimate search-token importance.

historical frames to the current frame, historical trajectories can provide useful priors for token importance estimation. Based on this observation, we reformulate token pruning in visual tracking as a trajectory-guided reinforcement learning decision-making problem. Given the recent K-frame historical target trajectory $\mathcal{H}_t = \{\mathbf{b}_{t-K}, \mathbf{b}_{t-K+1}, \ldots, \mathbf{b}_{t-1}\}$, a policy network samples a pruning action $\mathcal{A}_t$ from the trajectory state $\mathcal{S}_t$:

$$\mathcal{A}_t \sim \pi_\theta(\mathcal{A}_t \mid \mathcal{S}_t).$$

The action determines the motion prior, motion uncertainty, and token keep ratio for the current frame. Under this formulation, the token importance score is estimated not only from current-frame target responses, but also from historical trajectory priors. The policy is optimized by reward signals that balance tracking accuracy, target-token preservation, and computational cost. Thus, TGTP aims to learn a trajectory-aware token pruning policy that adaptively selects informative search tokens across frames, thereby reducing model FLOPs while maintaining tracking performance.

3.2 Trajectory-Guided Token Pruning

The proposed Trajectory-Guided Token Pruning (TGTP) is a general token pruning framework that can be integrated into existing visual trackers. As shown in Fig. 2, TGTP does not change the overall architecture of the original tracker, but prunes search-region tokens at selected network layers to reduce the number of tokens and computational cost in subsequent layers. Given the template tokens $\mathbf{Z}$ and the search tokens $\mathbf{X}_t$ in the t-th frame, the original tracker first performs template-search feature interaction in the backbone or encoder. When features arrive at a pruning layer, TGTP estimates the importance of each search token according to the template-search attention response at the current layer and the historical trajectory prior.

Specifically, the token importance estimation in TGTP consists of two complementary components. The first component is the current-frame template-search attention response $\mathbf{A}_t^\ell$, which measures the appearance correlation between search tokens and the target. The second is the trajectory-guided motion prior $\mathbf{P}_t^\ell$, which describes the possible spatial region where the target may appear in the current frame. This motion prior is predicted by the TGRL module from historical target trajectories. Before fusion, both score maps are normalized over search tokens to make their scales comparable: $\bar{\mathbf{A}}_t^\ell = \mathcal{N}(\mathbf{A}_t^\ell)$ and $\bar{\mathbf{P}}_t^\ell = \mathcal{N}(\mathbf{P}_t^\ell)$, where $\mathcal{N}(\cdot)$ denotes min-max normalization. TGTP fuses these two normalized components to obtain a trajectory-enhanced token importance map:

$$\widetilde{\mathbf{A}}_t^\ell = \alpha\bar{\mathbf{A}}_t^\ell + \beta\bar{\mathbf{P}}_t^\ell,$$

where α and β are the fusion weights.

In the token pruning stage, TGTP determines the number of search tokens to retain according to the keep ratio predicted by TGRL, and selects top-ranked search tokens based on $\widetilde{\mathbf{A}}_t^\ell$ to form the retained token set. The retained search tokens are then fed into subsequent network layers, while the unretained tokens no longer participate in later computation. In this way, TGTP reduces the number of tokens and FLOPs in subsequent layers. It is worth noting that TGTP only prunes search-region tokens, while template tokens are always preserved to maintain stable target representation. Overall, TGTP extends token pruning in visual tracking from a static selection process mainly based on

current-frame responses to a trajectory-guided adaptive pruning process. In TGTP, TGRL is responsible for predicting the motion prior and keep ratio from historical trajectories. The detailed state construction, action definition, and reward design of TGRL are introduced in the next subsection.

3.3 Trajectory-Guided Reinforcement Learning

The Trajectory-Guided Reinforcement Learning (TGRL) module is the core of TGTP, which learns pruning actions from historical target trajectories. As illustrated in Fig. 2, TGRL consists of three components: State Construction, Trajectory Policy Network, and Reward Evaluation. Given the recent K-frame historical trajectory, TGRL first constructs a trajectory state. Then, the Trajectory Policy Network predicts the motion offset, motion uncertainty, and token keep ratio for the current frame. During training, the policy is optimized by reward signals that balance tracking accuracy, target-token preservation, and computational cost.

State Construction. For the t-th frame, the historical target trajectory is denoted as $\mathcal{H}_t = \{\mathbf{b}_{t-K}, \mathbf{b}_{t-K+1}, \ldots, \mathbf{b}_{t-1}\}$. The target box in the i-th frame is represented as

$$\mathbf{b}_i = (\mathbf{c}_i, \mathbf{r}_i), \qquad \mathbf{c}_i = (x_i, y_i), \qquad \mathbf{r}_i = (w_i, h_i),$$

where $\mathbf{c}_i$ denotes the target center and $\mathbf{r}_i$ denotes the target scale. To model short-term target motion and scale variation, we further define

$$\Delta\mathbf{c}_i = \mathbf{c}_i - \mathbf{c}_{i-1}, \qquad \Delta\mathbf{r}_i = \mathbf{r}_i - \mathbf{r}_{i-1}.$$

The trajectory state is then constructed as

$$\mathcal{S}_t = \left[\mathbf{c}_{t-K:t-1}, \mathbf{r}_{t-K:t-1}, \Delta\mathbf{c}_{t-K+1:t-1}, \Delta\mathbf{r}_{t-K+1:t-1}\right].$$

All box-related quantities are represented in the normalized search-region coordinate system. This state encodes historical target location, scale, center displacement, and scale variation, enabling the policy network to capture short-term target dynamics from historical frames to the current frame.

Trajectory Policy Network. Given the state $\mathcal{S}_t$, the Trajectory Policy Network models a hybrid pruning policy, which consists of a continuous motion branch and a discrete keep-ratio branch:

$$\pi_\theta(\mathcal{A}_t|\mathcal{S}_t) = \pi_\theta^m(\boldsymbol{\eta}_t|\mathcal{S}_t)\pi_\theta^\gamma(k_t|\mathcal{S}_t),$$

where π_θ^m denotes the continuous motion policy, π_θ^γ denotes the discrete keep-ratio policy, $\boldsymbol{\eta}_t$ is the sampled motion variable, and k_t is the selected keep-ratio index. This formulation matches the mixed action space of TGRL, where a raw motion variable is sampled from a continuous Gaussian policy and the keep-ratio index is sampled from a categorical policy. For the continuous motion branch, the policy network predicts the mean of a Gaussian distribution:

$$\boldsymbol{\eta}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \mathrm{diag}(\boldsymbol{\sigma}_a^2)\right), \qquad \boldsymbol{\mu}_t = g_\theta^m(\mathcal{S}_t),$$

where $\boldsymbol{\eta}_t \in \mathbb{R}^4$ is the sampled raw motion variable. It is then decoded into the target-center offset and motion uncertainty:

$$(\Delta\hat{x}_t, \Delta\hat{y}_t) = \delta_{\max}\tanh(\boldsymbol{\eta}_{t,1:2}),$$
$$(\hat{\sigma}_{x,t}, \hat{\sigma}_{y,t}) = \sigma_{\min} + \mathrm{Sigmoid}(\boldsymbol{\eta}_{t,3:4})(\sigma_{\max} - \sigma_{\min}).$$

The action is defined as

$$\mathcal{A}_t = (\Delta\hat{x}_t, \Delta\hat{y}_t, \hat{\sigma}_{x,t}, \hat{\sigma}_{y,t}, \gamma_t).$$

Here, $\Delta\hat{x}_t$ and $\Delta\hat{y}_t$ denote the predicted target-center offset relative to the previous frame, $\hat{\sigma}_{x,t}$ and $\hat{\sigma}_{y,t}$ denote the motion uncertainty along the horizontal and vertical directions, and γ_t denotes the token keep ratio for the current frame. The keep ratio is selected from a discrete candidate set $\Gamma = \{\gamma^{(1)}, \gamma^{(2)}, \ldots, \gamma^{(C)}\}$. The policy network outputs a categorical distribution over Γ:

$$\mathbf{p}_t^\gamma = \mathrm{Softmax}\left(g_\theta^\gamma(\mathcal{S}_t)\right),$$

where $g_\theta^\gamma(\cdot)$ is the keep-ratio prediction head. During training, a keep-ratio index is sampled as

$$k_t \sim \mathrm{Cat}(\mathbf{p}_t^\gamma), \qquad \gamma_t = \Gamma[k_t],$$

while during inference, the keep ratio with the maximum probability is selected. Here, $\mathrm{Cat}(\cdot)$ denotes the categorical distribution. This enables frame-wise adaptive pruning. The predicted target center is obtained by

$$\hat{\mathbf{c}}_t = \mathbf{c}_{t-1} + (\Delta\hat{x}_t, \Delta\hat{y}_t).$$

Let $\mathbf{p}_i^\ell = (u_i^\ell, v_i^\ell)$ denote the spatial center of the i-th search token at layer ℓ. Based on the predicted target center and motion uncertainty, the motion prior for this search token is computed as

$$\mathbf{P}_{t,i}^\ell = \exp\left(-\frac{(u_i^\ell - \hat{x}_t)^2}{2\hat{\sigma}_{x,t}^2} - \frac{(v_i^\ell - \hat{y}_t)^2}{2\hat{\sigma}_{y,t}^2}\right),$$

where $\hat{\mathbf{c}}_t = (\hat{x}_t, \hat{y}_t)$. The motion prior map is denoted as $\mathbf{P}_t^\ell = \{\mathbf{P}_{t,i}^\ell\}_{i=1}^{N_{t,\ell}^x}$, while γ_t determines the proportion of search tokens to retain. Compared with directly predicting the absolute target position, predicting the relative offset better matches the motion continuity assumption in visual tracking and is easier to learn from short-term historical trajectories.

Reward Evaluation. During training, TGRL evaluates the quality of the pruning action through a reward function. The reward jointly considers tracking accuracy, computational cost, motion uncertainty, and target-token preservation:

$$\mathcal{R}_t = \lambda_{\text{iou}}\text{IoU}_t - \lambda_{\text{flops}}\text{FLOPs}_t - \lambda_\sigma \Omega_{\sigma,t} + \lambda_{\text{keep}}\rho_t^{\text{keep}}.$$

Here, IoU_t measures the overlap between the predicted box and the ground-truth box, encouraging accurate tracking. FLOPs_t denotes the normalized computational cost estimated from the retained search tokens in subsequent layers, encouraging the policy to reduce redundant computation. $\Omega_{\sigma,t}$ is the motion uncertainty regularization term:

$$\Omega_{\sigma,t} = \hat{\sigma}_{x,t}^2 + \hat{\sigma}_{y,t}^2.$$

The target-token preservation term ρ_t^{keep} is defined as

$$\rho_t^{\text{keep}} = \frac{1}{|L_p|}\sum_{\ell \in L_p} \frac{|\mathcal{M}_t^\ell \cap \mathcal{G}_t^\ell|}{|\mathcal{G}_t^\ell|},$$

where $\mathcal{M}_t^\ell$ denotes the retained token set and $\mathcal{G}_t^\ell$ denotes the target-related token set at pruning layer ℓ. Specifically, $\mathcal{G}_t^\ell$ is obtained by projecting the ground-truth box $\mathbf{b}_t^{\text{gt}}$ onto the search-token grid:

$$\mathcal{G}_t^\ell = \{i \mid \mathbf{p}_i^\ell \in \mathbf{b}_t^{\text{gt}}\}.$$

If no token center falls inside the ground-truth box, the token closest to the ground-truth target center is included to avoid an empty target-token set. This term encourages the policy to preserve target-related tokens during pruning. Overall, IoU_t and ρ_t^{keep} encourage reliable tracking, FLOPs_t encourages lower computational cost, and $\Omega_{\sigma,t}$ prevents the motion prior from becoming overly dispersed. With this reward design, TGRL learns a trajectory-aware pruning policy and provides TGTP with the layer-wise motion prior $\mathbf{P}_t^\ell$ and an adaptive keep ratio γ_t. Since γ_t is predicted from the historical trajectory state, TGTP can dynamically adjust the number of retained search tokens according to target motion and scene variations.

3.4 Training Objective and Inference

During training, TGTP is jointly optimized with the original tracker. The original tracker keeps its own tracking loss for target localization, such as classification loss, regression loss, or IoU-related loss. In addition, TGTP introduces the reinforcement learning loss of TGRL to optimize the Trajectory Policy Network, enabling it to learn suitable motion priors and adaptive keep ratios. The overall training objective is formulated as

$$\mathcal{L} = \mathcal{L}_{\text{track}} + \lambda_{\text{RL}}\mathcal{L}_{\text{RL}},$$

where $\mathcal{L}_{\text{track}}$ denotes the original tracking loss, $\mathcal{L}_{\text{RL}}$ denotes the reinforcement learning loss of TGRL, and λ_{RL} balances the two losses.

For TGRL, we adopt an actor-critic formulation to optimize the policy network. The Trajectory Policy Network predicts the pruning action $\mathcal{A}_t$ from the state $\mathcal{S}_t$, while the value branch estimates the expected return of the current state, denoted as $V(\mathcal{S}_t)$. Based on the reward $\mathcal{R}_t$ and the value prediction, the advantage is computed as

$$\mathcal{D}_t = \mathcal{R}_t - V(\mathcal{S}_t),$$

Table 1: State-of-the-art comparisons of real-time single-modal tracking methods on five widely used tracking datasets. The $*$ denotes only training with the GOT-10k training set. The **bold** and *underline* fonts represent the **best** and *second-best* tracking results, respectively.

Single-Modal Tracking: Real-Time RGB Tracking													
Method	Source	LaSOT			GOT-10k*			TrackingNet			LaSOT_{ext}	NFS30	FLOPs
		AUC	PN	P	AO	SR50	SR75	AUC	PN	P	AUC	AUC	(G)↓
FERMT Zheng et al. (2024)	ECCV24	65.1	74.6	69.1	69.6	80.1	63.2	80.8	85.9	78.1	*46.1*	*65.1*	2.39 G
DyHiT Kang et al. (2025b)	IJCV25	62.4	70.1	64.0	62.9	71.8	55.2	77.9	82.2	74.0	42.1	61.6	4.34 G
AsymTrack-B Zhu et al. (2025)	AAAI25	64.7	73.0	67.8	67.7	76.6	61.4	80.0	84.5	77.4	44.6	64.4	1.80 G
MST Wang et al. (2025)	MM25	*65.8*	75.2	*70.1*	69.6	79.8	*64.1*	*81.0*	*86.1*	*78.7*	45.1	**65.4**	2.44 G
UETrack-T Kang et al. (2026)	CVPR26	63.4	72.5	65.1	65.3	75.1	58.4	78.9	83.8	74.8	42.2	-	1.80 G
SpikeTrack-S256 Zhang et al. (2026)	CVPR26	64.5	*76.0*	69.0	67.8	79.9	61.7	78.7	84.6	75.5	43.9	-	-
FARTrack-nano Wang et al. (2026)	ICLR26	61.3	69.7	64.1	*69.9*	*81.2*	61.4	79.1	84.5	75.6	43.8	65.1	2.99 G
Average number of search region tokens: 256 → 175, saving 31.6%													
FERMT Zheng et al. (2024)	ECCV24	65.1	74.6	69.1	69.6	80.1	63.2	80.8	85.9	78.1	46.1	65.1	2.39 G
+ TGTP	**Ours**	**66.7**	**76.6**	**71.4**	**71.3**	**81.9**	**66.8**	**81.4**	**86.6**	**79.3**	**46.6**	**65.4**	2.02 G
Δ		*1.6*↑	*2.0*↑	*2.3*↑	*1.7*↑	*1.8*↑	*3.6*↑	*0.6*↑	*0.7*↑	*1.2*↑	*0.5*↑	*0.3*↑	*15.4%*↓

Table 2: State-of-the-art comparisons of non-real-time single-modal methods on five datasets.

Single-Modal Tracking: Non-Real-Time RGB Tracking														
Method	Source	GOT-10k*			LaSOT			LaSOT_{ext}			NFS30		TNL2K		FLOPs
		AO	SR50	SR75	AUC	PN	P	AUC	PN	P	AUC	P	AUC	P	(G)↓
RSTrack-256 Zeng et al. (2025)	MM25	76.6	86.4	76.9	73.4	84.1	81.7	52.7	63.8	60.3	-	-	60.5	-	57.0 G
MCITrack-B224 Kang et al. (2025a)	AAAI25	77.9	88.2	76.8	**75.3**	**85.6**	**83.3**	*54.6*	*65.7*	62.1	70.6	*85.4*	*82.9*	*66.7*	38.5 G
SPMTrack-B Cai et al. (2025)	CVPR25	76.5	85.9	76.3	74.9	84.0	81.7	-	-	-	67.4	**85.7**	62.0	*66.7*	-
DreamTrack-256 Guo et al. (2025)	CVPR25	77.5	88.2	76.8	73.8	83.4	80.6	53.1	64.1	59.8	65.4	-	60.4	63.2	37.6 G
MoDTrack-256 Yang et al. (2026a)	AAAI26	74.0	84.1	71.9	72.9	82.6	79.1	52.4	62.9	60.5	-	-	59.7	62.3	35.2 G
MUTrack-B256 Wu et al. (2026b)	AAAI26	78.1	88.3	77.5	72.3	82.9	79.5	51.2	62.5	58.8	69.7	-	-	-	39.2 G
SpikeTrack-B256 Zhang et al. (2026)	CVPR26	72.2	83.9	67.7	67.1	77.7	72.5	46.7	57.6	52.9	-	-	53.3	-	-
PNTrack-256 Zheng et al. (2026)	CVPR26	69.0	79.1	62.7	65.9	76.3	70.2	47.0	58.8	53.1	-	-	53.4	-	-
DTPTrack-B224 Huang et al. (2026)	CVPR26	76.5	87.6	74.9	74.3	83.6	80.7	-	-	-	-	-	60.3	-	53.8 G
ETCTrack-B224 Wu et al. (2026c)	CVPR26	*79.2*	*90.2*	78.6	74.9	85.1	82.7	*54.6*	**65.9**	*62.4*	**71.3**	-	61.3	65.9	35.9 G
Average number of search region tokens: 196 → 135, saving 31.1%															
MCITrack-B224 Kang et al. (2025a)	AAAI25	77.9	88.2	76.8	75.3	85.6	83.3	54.6	65.7	62.1	70.6	85.4	62.9	66.7	38.5 G
+ TGTP	**Ours**	**80.6**	**90.9**	**80.6**	*75.1*	*85.4*	*82.9*	**54.7**	*65.7*	**62.5**	*70.7*	**85.7**	**63.0**	**67.2**	34.5 G
Δ		*2.7*↑	*2.7*↑	*3.8*↑	*0.2*↓	*0.2*↓	*0.4*↓	*0.1*↑	*0.0*↑	*0.4*↑	*0.1*↑	*0.3*↑	*0.1*↑	*0.5*↑	*10.4%*↓

where $\mathcal{D}_t$ measures whether the sampled action performs better or worse than the expected return. When $\mathcal{D}_t > 0$, the sampled action receives a higher reward than expected and its probability should be increased; otherwise, its probability should be reduced. The policy loss is defined as

$$\mathcal{L}_{\text{policy}} = -\left[\log \pi_\theta^m(\boldsymbol{\eta}_t|\mathcal{S}_t) + \log \pi_\theta^\gamma(k_t|\mathcal{S}_t)\right]\mathcal{D}_t.$$

Meanwhile, the value branch is trained by regressing the reward:

$$\mathcal{L}_{\text{value}} = \left(V(\mathcal{S}_t) - \mathcal{R}_t\right)^2.$$

To avoid premature convergence to a single action, we further introduce entropy regularization to encourage exploration during training. The reinforcement learning loss of TGRL is then written as

$$\mathcal{L}_{\text{RL}} = \mathcal{L}_{\text{policy}} + \lambda_v \mathcal{L}_{\text{value}} - \lambda_e \mathcal{E}(\pi_\theta),$$

where λ_v and λ_e are loss weights, and $\mathcal{E}(\pi_\theta)$ is the entropy of the policy distribution. During inference, TGTP does not compute rewards or update the policy network. TGRL only performs a forward pass based on the historical trajectory state to predict the pruning action. Then, TGTP computes the token importance map using the predicted motion prior and current-layer template-search attention, and performs token pruning at selected pruning layers. Since the removed search tokens no longer participate in subsequent network layers, TGTP reduces model FLOPs, while template tokens are always preserved to maintain stable target representation.

4 Experiments

4.1 Implementation Details

Model. TGTP is integrated as a lightweight plug-in while keeping the original settings of each baseline tracker unchanged. For ViT-based trackers, token pruning is performed at layers 3, 6, and 9 unless otherwise specified, and the trajectory state uses $K = 3$ historical frames. The Trajectory Policy Network uses a hidden dimension of 128. The candidate keep-ratio set is $\{0.7, 0.75, 0.8\}$. For token importance fusion, the weights of the attention response and trajectory prior are set to 1.0 and 0.5, respectively. In the reward function, we set $\lambda_{\text{iou}} = 1.0$, $\lambda_{\text{flops}} = 0.05$, $\lambda_\sigma = 0.02$, and $\lambda_{\text{keep}} = 0.2$. For the actor-critic loss, we set $\lambda_{\text{RL}} = 0.5$, $\lambda_v = 0.2$, and $\lambda_e = 0.01$.

Table 3: State-of-the-art comparisons of multi-modal tracking methods on four datasets.

Multi-Modal Tracking: RGB-Thermal/ Event/ Depth Tracking										
		RGB-T				**RGB-E**		**RGB-D**		
Method	Source	RGBT234		LasHeR		VisEvent		VOT-RGBD22		FLOPs
		MSR	MPR	AUC	P	AUC	P	EAO	Rob	(G)↓
XTrack-B Tan et al. (2025)	ICCV25	64.9	87.4	55.7	69.1	60.9	77.5	74.0	88.8	-
SMSTracker Chan et al. (2025)	ICCV25	64.5	86.9	56.0	70.3	60.4	76.3	74.8	89.7	-
SUTrack-B224 Chen et al. (2025)	AAAI25	69.5	92.2	59.9	74.5	62.7	79.9	76.5	91.8	19.2 G
OneTrackerV2-B224 Hong et al. (2026)	ICML26	69.5	91.3	60.6	75.5	63.0	79.8	-	-	23.8 G
SEATrack Su et al. (2026)	CVPR26	63.9	87.8	57.3	71.6	60.3	77.1	73.6	88.4	-
UETrack-B Kang et al. (2026)	CVPR26	64.2	86.7	55.5	69.1	59.2	76.2	68.3	-	3.2 G
UTPTrack-224 Wu et al. (2026a)	CVPR26	*70.4*	92.9	59.2	73.9	62.6	80.0	76.4	92.1	16.0 G
TGTrack-B256 Geng et al. (2026)	CVPR26	69.9	*93.7*	*61.7*	*77.4*	*63.9*	*81.4*	**77.4**	*93.0*	44.0 G
Average number of search region tokens: 196 → 110, saving 43.9%										
SUTrack-B224 Chen et al. (2025)	AAAI25	69.5	92.2	59.9	74.5	62.7	79.9	76.5	91.8	19.2 G
+ TGTP	**Ours**	**71.1**	**95.0**	**62.0**	**78.3**	**64.4**	**82.1**	*77.3*	**93.6**	12.9 G
Δ		*1.6*↑	*2.8*↑	*2.1*↑	*3.8*↑	*1.7*↑	*2.2*↑	*0.8*↑	*1.8*↑	*32.8%*↓

Table 4: Comparison with other token pruning methods, using the same pruning ratio.

Method	Source	LaSOT			GOT-10k*			LaSOT$_{ext}$			FLOPs
		AUC	PN	P	AO	SR50	SR75	AUC	PN	P	(G)↓
OSTrack-EViT Liang et al. (2022)	ICLR21	67.8	76.8	*73.4*	72.9	82.8	70.4	*47.1*	*56.6*	*52.8*	-
OSTrack-ToMe Bolya et al. (2022)	ICLR22	*68.2*	*77.1*	73.2	*74.9*	**85.1**	**73.6**	46.5	56.0	52.2	-
OSTrack-UTP Wu et al. (2026a)	CVPR26	67.6	76.3	72.9	74.8	**85.1**	*73.5*	**47.3**	**57.2**	**53.0**	24.0 G
OSTrack-TGTP	**Ours**	**68.6**	**78.1**	**74.5**	**75.0**	**85.1**	73.4	47.0	**57.2**	52.7	22.5 G

Training. As shown in Fig. 2, during fine-tuning, we optimize only the TGRL module and the tracking head, while freezing the remaining parameters of the original tracker. All models are trained for 30 epochs using AdamW with an initial learning rate of 1×10^{-4}. The learning rate is decayed by a factor of 0.1 at the 20-th epoch. For fair comparison, the training datasets, sampling strategy, and number of training samples are kept the same as those of the corresponding baseline trackers.

4.2 Comparison with State-of-the-Art Methods

To validate the generality of TGTP, we integrate it into different visual trackers and evaluate it on both single-modal and multi-modal tracking tasks. For single-modal tracking, we consider real-time and non-real-time RGB trackers on LaSOT Fan et al. (2019), GOT-10k Huang et al. (2019), TrackingNet Muller et al. (2018), LaSOT$_{ext}$ Fan et al. (2021), NFS30 Kiani Galoogahi et al. (2017), and TNL2K Wang et al. (2021), with results reported in Tab. 1 and Tab. 2. For multi-modal tracking, we evaluate RGB-T, RGB-E, and RGB-D trackers on RGBT234 Li et al. (2019), LasHeR Li et al. (2021), VisEvent Wang et al. (2023b), and VOT-RGBD22 Kristan et al. (2022), as shown in Tab. 3.

As illustrated in Tab. 1, TGTP improves real-time single-modal tracking while reducing computational cost. When integrated into FERMT Zheng et al. (2024), TGTP reduces FLOPs from 2.39G to 2.02G, with a 15.4% reduction. Meanwhile, it improves AUC from 65.1 to 66.7 on LaSOT, AO from 69.6 to 71.3 on GOT-10k, and AUC from 80.8 to 81.4 on TrackingNet. It also brings AUC gains of 1.2 and 0.3 on LaSOT$_{ext}$ and NFS30, respectively. For non-real-time single-modal tracking, Tab. 2 shows that TGTP also generalizes well to strong baseline trackers. When integrated into MCITrack-B224 Kang et al. (2025a), TGTP reduces FLOPs from 38.5G to 34.5G, with a 10.4% reduction. It improves AO from 77.9 to 80.6 on GOT-10k, AUC from 54.6 to 54.7 on LaSOT$_{ext}$, AUC from 70.6 to 70.7 on NFS30, and brings gains of 0.1 AUC and 0.5 P on TNL2K. Although some metrics on LaSOT slightly decrease, the overall results indicate that TGTP can further reduce computational cost on strong trackers while maintaining competitive tracking performance.

As shown in Tab. 3, TGTP also adapts well to multi-modal RGB-X tracking. When integrated into SUTrack-B224 Chen et al. (2025), TGTP reduces FLOPs from 19.2G to 12.9G, achieving a 32.8% reduction. Meanwhile, it improves MSR from 69.5 to 71.1 on RGBT234 and AUC from 59.9 to 62.0 on LasHeR. On the RGB-E dataset VisEvent, AUC and P are improved from 62.7 and 79.9 to 64.4 and 82.1, respectively. On the RGB-D dataset VOT-RGBD22, Rob is improved from 91.8 to 93.6. These results demonstrate that TGTP is not limited to a specific modality or tracker architecture. Instead, it consistently reduces redundant search tokens and model FLOPs across single-modal and multi-modal tracking scenarios while maintaining or improving tracking performance.

To further verify the effectiveness of TGTP for token pruning, we compare it with EViT Liang et al. (2022), ToMe Bolya et al. (2022), and UTP Wu et al. (2026a) under the same pruning ratio, as shown

Table 5: Ablation studies on the number of historical frames, score fusion weights, candidate keep ratio sets, and reward components.

K	UAVDT AUC	UAVDT P
1	61.0	84.1
2	61.6	84.7
3	**62.4**	**85.5**
4	61.6	85.3

α	β	UAVDT AUC	UAVDT P
1.0	0.5	**62.4**	**85.5**
0.5	1.0	61.2	84.9
1.0	1.0	61.6	85.3
1.0	0.2	61.1	84.6

Candidate keep ratio set	UAVDT AUC	UAVDT P
{0.65,0.7,0.75}	61.8	85.1
{0.7,0.75,0.8}	**62.4**	**85.5**
{0.7,0.8,0.9}	61.1	84.5
{0.6,0.7,0.8,0.9}	60.7	83.8

Reward component	UAVDT AUC	UAVDT P
IoU reward	60.9	84.5
+ FLOPs penalty	61.4	84.7
+ Sigma regularization	61.7	85.2
+ Target-token reward	**62.4**	**85.5**

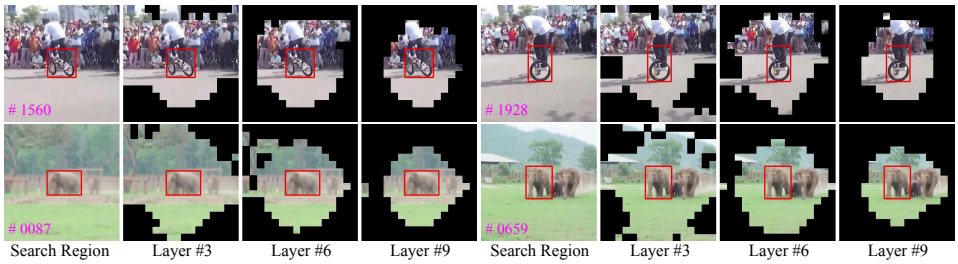


Figure 3: Visualization of trajectory-guided token pruning.

in Tab. 4. OSTrack-TGTP achieves 68.6 AUC, 78.1 PN, and 74.5 P on LaSOT, outperforming the compared pruning methods. On GOT-10k, TGTP obtains 75.0 AO and 85.1 SR50, achieving competitive performance. On LaSOT$_{ext}$, TGTP also shows comparable results to the best-performing pruning baseline. Meanwhile, TGTP reduces FLOPs to 22.5G, lower than 24.0G of OSTrack-UTP. These results indicate that TGTP more effectively preserves target-related tokens and achieves a better balance between tracking performance and computational cost under the same pruning setting.

4.3 Ablation Study

To analyze the influence of key designs in TGTP, we conduct ablation studies on UAVDT Li & Yeung (2017), as reported in Tab. 5. First, the number of historical frames K clearly affects tracking performance. When $K = 1$, the model only uses limited temporal information and obtains 61.0 AUC. The performance improves as more historical frames are introduced, reaching the best result at $K = 3$ with 62.4 AUC and 85.5 P. When K increases to 4, the performance slightly drops, indicating that overly long histories may introduce outdated motion cues. Second, the fusion weights between the attention response and trajectory prior are also important. The setting $\alpha = 1.0, \beta = 0.5$ achieves the best performance, suggesting that trajectory priors provide useful complementary guidance while current-frame attention remains the main cue. Third, for the candidate keep-ratio set, $\{0.7, 0.75, 0.8\}$ performs best, while overly aggressive or overly broad candidates lead to lower performance. Finally, adding reward components progressively improves AUC from 60.9 with only the IoU reward to 62.4 with the full reward, confirming the effectiveness of the FLOPs penalty, sigma regularization, and target-token reward.

In addition, we visualize the token pruning process of TGTP in Fig. 3. As the network goes deeper, TGTP progressively removes a large number of background tokens while preserving search tokens around the target region. This shows that the trajectory prior provides effective spatial guidance for token importance estimation, making pruning more focused on likely target regions and helping reduce computation while maintaining target representation.

5 Conclusion

In this paper, we propose Trajectory-Guided Token Pruning (TGTP), a general token pruning framework for visual tracking. TGTP exploits the temporal continuity of tracking and formulates token pruning as a trajectory-guided reinforcement learning problem. By constructing states from historical target trajectories, the proposed TGRL module learns motion priors, motion uncertainty, and adaptive keep ratios to guide token selection. As a lightweight plug-in module, TGTP can be integrated into different single-modal and multi-modal trackers without redesigning their original architectures. Extensive experiments show that TGTP consistently reduces search tokens and model FLOPs while maintaining or improving tracking performance, demonstrating the effectiveness and generality of trajectory-guided reinforcement learning for computation-efficient visual tracking.

AI USE STATEMENT

We used generative AI tools only to polish the writing of this paper, including improving grammar, clarity, and readability. We did not use generative AI tools to generate research ideas, formulate hypotheses, design the methodology or experiments, implement the proposed method, analyze data, interpret results, create figures, or derive mathematical claims. All technical content, experimental results, and scientific claims were produced and verified by the authors. The authors reviewed and edited all AI-assisted text and take full responsibility for the final content of this work.

REFERENCES

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Wenrui Cai, Qingjie Liu, and Yunhong Wang. Spmtrack: Spatio-temporal parameter-efficient fine-tuning with mixture of experts for scalable visual tracking. In *CVPR*, pp. 16871–16881, 2025.

Sixian Chan, Zedong Li, Wenhao Li, Shijian Lu, Chunhua Shen, and Xiaoqin Zhang. Smstracker: Tri-path score mask sigma fusion for multi-modal tracking. In *ICCV*, pp. 4766–4775, 2025.

Xin Chen, Ben Kang, Wanting Geng, Jiawen Zhu, Yi Liu, Dong Wang, and Huchuan Lu. Sutrack: Towards simple and unified single object tracking. In *AAAI*, volume 39, pp. 2239–2247, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, pp. 1–21, 2021.

Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, pp. 5374–5383, 2019.

Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, et al. Lasot: A high-quality large-scale single object tracking benchmark. *IJCV*, 129(2):439–461, 2021.

Wanting Geng, Xin Chen, Chuanyu Sun, Jie Zhao, Ben Kang, Dong Wang, and Huchuan Lu. Tg-track: Temporal generative learning for unified single object tracking. In *CVPR*, pp. 28134–28144, 2026.

Mingzhe Guo, Weiping Tan, Wenyu Ran, Liping Jing, and Zhipeng Zhang. Dreamtrack: Dreaming the future for multimodal visual object tracking. In *CVPR*, pp. 7201–7210, 2025.

Lingyi Hong, Jinglun Li, Xinyu Zhou, Kaixun Jiang, Pinxue Guo, Zhaoyu Chen, Runze Li, Xingdong Sheng, and Wenqiang Zhang. Unified multimodal visual tracking with dual mixture-of-experts. *arXiv preprint arXiv:2605.03716*, 2026.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE TPAMI*, 43(5):1562–1577, 2019.

Yuqing Huang, Liting Lin, Weijun Zhuang, Zhenyu He, and Xin Li. Drift-resilient temporal priors for visual tracking. In *CVPR*, pp. 6847–6856, 2026.

Ben Kang, Xin Chen, Simiao Lai, Yang Liu, Yi Liu, and Dong Wang. Exploring enhanced contextual information for video-level object tracking. In *AAAI*, volume 39, pp. 4194–4202, 2025a.

Ben Kang, Xin Chen, Jie Zhao, Chunjuan Bo, Dong Wang, and Huchuan Lu. Exploiting lightweight hierarchical vit and dynamic framework for efficient visual tracking. *IJCV*, 133(10):6689–6711, 2025b.

Ben Kang, Jie Zhao, Xin Chen, Wanting Geng, Bin Zhang, Lu Zhang, Dong Wang, and Huchuan Lu. Uetrack: A unified and efficient framework for single object tracking. In *CVPR*, pp. 20890–20901, 2026.

Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, pp. 1125–1134, 2017.

Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Hyung Jin Chang, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, et al. The tenth visual object tracking vot2022 challenge results. In *ECCV*, pp. 431–460. Springer, 2022.

Chenglong Li, Xinyan Liang, Yijuan Lu, Nan Zhao, and Jin Tang. Rgb-t object tracking: Benchmark and baseline. *PR*, 96:106977, 2019.

Chenglong Li, Wanlin Xue, Yaqing Jia, Zhichen Qu, Bin Luo, Jin Tang, and Dengdi Sun. Lasher: A large-scale high-diversity benchmark for rgbt tracking. *IEEE TIP*, 31:392–404, 2021.

Siyi Li and Dit-Yan Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *AAAI*, volume 31, 2017.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, pp. 300–317, 2018.

Liangtao Shi, Bineng Zhong, Qihua Liang, Ning Li, Shengping Zhang, and Xianxian Li. Explicit visual prompts for visual object tracking. In *AAAI*, pp. 4838–4846, 2024.

Junbin Su, Ziteng Xue, Shihui Zhang, Kun Chen, Weiming Hu, and Zhipeng Zhang. Seatrack: Simple, efficient, and adaptive multimodal tracker. In *CVPR*, pp. 28679–28689, 2026.

Yuedong Tan, Zongwei Wu, Yuqian Fu, Zhuyun Zhou, Guolei Sun, Eduard Zamfir, Chao Ma, Danda Paudel, Luc Van Gool, and Radu Timofte. Xtrack: Multimodal training boosts rgb-x video object trackers. In *ICCV*, pp. 5734–5744, 2025.

Yunjie Tian, Lingxi Xie, Zhaozhi Wang, Longhui Wei, Xiaopeng Zhang, Jianbin Jiao, Yaowei Wang, Qi Tian, and Qixiang Ye. Integrally pre-trained transformer pyramid networks. In *CVPR*, pp. 18610–18620, 2023.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pp. 10347–10357, 2021.

Guijie Wang, Tong Lin, Yifan Bai, Anjia Cao, Shiyi Liang, Wangbo Zhao, and Xing Wei. Fartrack: Fast autoregressive visual tracking with high performance. *arXiv preprint arXiv:2602.03214*, 2026.

Shaoru Wang, Jin Gao, Zeming Li, Xiaoqin Zhang, and Weiming Hu. A closer look at self-supervised lightweight vision transformers. In *International Conference on Machine Learning*, pp. 35624–35641. PMLR, 2023a.

Shilei Wang, Gong Cheng, Pujian Lai, Dong Gao, and Junwei Han. Multi-state tracker: Enhancing efficient object tracking via multi-state specialization and interaction. In *ACMMM*, pp. 4087–4096, 2025.

Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *CVPR*, pp. 13763–13773, 2021.

Xiao Wang, Jianing Li, Lin Zhu, Zhipeng Zhang, Zhe Chen, Xin Li, Yaowei Wang, Yonghong Tian, and Feng Wu. Visevent: Reliable object tracking via collaboration of frame and event flows. *IEEE Transactions on Cybernetics*, 54(3):1997–2010, 2023b.

Hao Wu, Xudong Wang, Jialiang Zhang, Junlong Tong, Xinghao Chen, Junyan Lin, Yunpu Ma, and Xiaoyu Shen. Utptrack: Towards simple and unified token pruning for visual tracking. In *CVPR*, pp. 20963–20972, 2026a.

Weijing Wu, Qihua Liang, Bineng Zhong, Xiaohu Tang, Yufei Tan, Ning Li, and Yuanliang Xue. Mutrack: A memory-aware unified representation framework for visual tracking. In *AAAI*, volume 40, pp. 10772–10780, 2026b.

Weijing Wu, Qihua Liang, Bineng Zhong, Haiying Xia, Zhiyi Mo, and Shuxiang Song. An efficient token compression framework for visual object tracking. In *CVPR*, pp. 6857–6867, 2026c.

You Wu, Xucheng Wang, Xiangyang Yang, Mengyuan Liu, Dan Zeng, Hengzhou Ye, and Shuiwang Li. Learning occlusion-robust vision transformers for real-time uav tracking. In *CVPR*, pp. 17103–17113, 2025.

Jinxia Xie, Bineng Zhong, Qihua Liang, Ning Li, Zhiyi Mo, and Shuxiang Song. Robust tracking via mamba-based context-aware token learning. In *AAAI*, volume 39, pp. 8727–8735, 2025.

Chaocan Xue, Bineng Zhong, Qihua Liang, Yaozong Zheng, Ning Li, Yuanliang Xue, and Shuxiang Song. Similarity-guided layer-adaptive vision transformer for uav tracking. In *CVPR*, pp. 6730–6740, 2025.

Hongtao Yang, Bineng Zhong, Qihua Liang, Xiantao Hu, Yufei Tan, Haiying Xia, and Shuxiang Song. Motion-aware object tracking via motion and geometry-aware cues. In *AAAI*, volume 40, pp. 11604–11612, 2026a.

Hongtao Yang, Bineng Zhong, Qihua Liang, Yaozong Zheng, Xiantao Hu, Yuanliang Xue, and Shuxiang Song. Dual-branch distilled transformer for efficient asymmetric uav tracking. In *CVPR*, pp. 13615–13625, 2026b.

Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *ECCV*, pp. 341–357, 2022.

Fansheng Zeng, Bineng Zhong, Haiying Xia, Yufei Tan, Xiantao Hu, Liangtao Shi, and Shuxiang Song. Explicit context reasoning with supervision for visual tracking. In *ACMMM*, pp. 8067–8076, 2025.

Qiuyang Zhang, Jiujun Cheng, Qichao Mao, Cong Liu, Yu Fang, Yuhong Li, Mengying Ge, and Shangce Gao. Spiketrack: A spike-driven framework for efficient visual tracking. In *CVPR*, pp. 6802–6811, 2026.

Xiaosong Zhang, Yunjie Tian, Lingxi Xie, Wei Huang, Qi Dai, Qixiang Ye, and Qi Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In *International Conference on Learning Representations*, pp. 1–14, 2023.

Jikai Zheng, Mingjiang Liang, Shaoli Huang, and Jifeng Ning. Exploring the feature extraction and relation modeling for light-weight transformer tracking. In *ECCV*, pp. 110–126, 2024.

Yaozong Zheng, Qihua Liang, Bineng Zhong, Shuimu Zeng, Yuanliang Xue, Ning Li, and Shuxiang Song. Boosting self-supervised tracking with contextual prompts and noise learning. In *CVPR*, pp. 35197–35206, 2026.

Jiawen Zhu, Huayi Tang, Xin Chen, Xinying Wang, Dong Wang, and Huchuan Lu. Two-stream beats one-stream: Asymmetric siamese network for efficient visual tracking. In *AAAI*, volume 39, pp. 10959–10967, 2025.

A Appendix

This appendix supplements the main paper with additional methodological details and experimental analyses. Specifically, we first elaborate the formulation of the Trajectory-Guided Reinforcement Learning (TGRL) module, including the construction of trajectory states, the definition of hybrid actions, the reward function, and the actor-critic optimization objective. We then provide implementation details, such as pruning-layer settings for different backbone networks and the hyperparameters used in TGTP. In addition, we present extended comparisons across diverse visual backbones, attribute-based analysis on LaSOT, and detailed ablation studies. We also provide qualitative tracking results and discuss the limitations of the current design.

A.1 Detailed Formulation of TGRL

In the main paper, we introduce the overall framework of TGTP and the role of TGRL in trajectory-guided token pruning. Due to the space limitation, only the main formulation and key procedures are presented in the main text. In this section, we provide a more detailed formulation of TGRL, including state construction, hybrid action space, policy distribution, motion-prior generation, reward design, and training objective. Overall, TGRL aims to exploit historical target trajectories to provide temporal priors for token importance estimation, while introducing only lightweight computational overhead.

Given the recent K target boxes before the t-th frame, the historical trajectory is denoted as

$$\mathcal{H}_t = \{\mathbf{b}_{t-K}, \mathbf{b}_{t-K+1}, \ldots, \mathbf{b}_{t-1}\}.$$

The target box in the i-th frame is represented as

$$\mathbf{b}_i = (\mathbf{c}_i, \mathbf{r}_i), \qquad \mathbf{c}_i = (x_i, y_i), \qquad \mathbf{r}_i = (w_i, h_i),$$

where $\mathbf{c}_i$ denotes the target center and $\mathbf{r}_i$ denotes the target scale. To describe short-term motion and scale variation, we further define

$$\Delta\mathbf{c}_i = \mathbf{c}_i - \mathbf{c}_{i-1}, \qquad \Delta\mathbf{r}_i = \mathbf{r}_i - \mathbf{r}_{i-1}.$$

The trajectory state is then constructed as

$$\mathcal{S}_t = \left[\mathbf{c}_{t-K:t-1}, \mathbf{r}_{t-K:t-1}, \Delta\mathbf{c}_{t-K+1:t-1}, \Delta\mathbf{r}_{t-K+1:t-1}\right].$$

All box-related quantities are represented in the normalized search-region coordinate system. This state explicitly encodes target location, scale, center displacement, and scale variation across recent frames. Such a design allows TGRL to capture short-term target dynamics using only historical tracking results, without introducing additional heavy visual features.

TGRL predicts both the motion prior and the token keep ratio. Therefore, the pruning action is defined as

$$\mathcal{A}_t = (\Delta\hat{x}_t, \Delta\hat{y}_t, \hat{\sigma}_{x,t}, \hat{\sigma}_{y,t}, \gamma_t).$$

Here, $\Delta\hat{x}_t$ and $\Delta\hat{y}_t$ denote the predicted target-center offset relative to the previous frame, $\hat{\sigma}_{x,t}$ and $\hat{\sigma}_{y,t}$ denote the motion uncertainty along the horizontal and vertical directions, and γ_t denotes the token keep ratio. This forms a hybrid action space: the motion offset and uncertainty are continuous variables, while the keep ratio is selected from a discrete candidate set. This design matches the nature of the problem, since target motion is continuous, whereas the pruning strength can be controlled by a small number of stable and interpretable keep-ratio candidates.

Accordingly, TGRL factorizes the policy into a continuous motion branch and a discrete keep-ratio branch:

$$\pi_\theta(\mathcal{A}_t|\mathcal{S}_t) = \pi_\theta^m(\boldsymbol{\eta}_t|\mathcal{S}_t)\pi_\theta^\gamma(k_t|\mathcal{S}_t),$$

where π_θ^m denotes the continuous motion policy, π_θ^γ denotes the discrete keep-ratio policy, $\boldsymbol{\eta}_t$ is the sampled raw motion variable, and k_t is the selected keep-ratio index. This factorization enables TGRL to handle motion modeling and pruning-ratio selection separately, making the policy easier to optimize.

For the continuous motion branch, the policy network predicts the mean of a Gaussian distribution:

$$\boldsymbol{\mu}_t = g_\theta^m(\mathcal{S}_t),$$

and samples the raw motion variable as

$$\boldsymbol{\eta}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_t, \mathrm{diag}(\boldsymbol{\sigma}_a^2)\right), \qquad \boldsymbol{\eta}_t \in \mathbb{R}^4.$$

The sampled variable is then decoded into the target-center offset and motion uncertainty:

$$(\Delta\hat{x}_t, \Delta\hat{y}_t) = \delta_{\max}\tanh(\boldsymbol{\eta}_{t,1:2}),$$

$$(\hat{\sigma}_{x,t}, \hat{\sigma}_{y,t}) = \sigma_{\min} + \mathrm{Sigmoid}(\boldsymbol{\eta}_{t,3:4})(\sigma_{\max} - \sigma_{\min}).$$

The $\tanh(\cdot)$ operation constrains the motion offset to avoid overly large position jumps, while the sigmoid mapping keeps the motion uncertainty within a reasonable range $[\sigma_{\min}, \sigma_{\max}]$. This improves the stability of the motion prior, especially under fast motion or temporary occlusion.

For the discrete keep-ratio branch, we define the candidate keep-ratio set as

$$\Gamma = \{\gamma^{(1)}, \gamma^{(2)}, \ldots, \gamma^{(C)}\}.$$

The policy network predicts a categorical distribution over the candidates:

$$\mathbf{p}_t^{\gamma} = \mathrm{Softmax}\left(g_\theta^{\gamma}(\mathcal{S}_t)\right).$$

During training, the keep-ratio index is sampled by

$$k_t \sim \mathrm{Cat}(\mathbf{p}_t^{\gamma}), \qquad \gamma_t = \Gamma[k_t].$$

During inference, we use a deterministic strategy:

$$k_t = \arg\max_c \mathbf{p}_{t,c}^{\gamma}, \qquad \gamma_t = \Gamma[k_t].$$

Using a discrete candidate set avoids unstable continuous variations of the pruning ratio and provides direct control over the computational cost. Compared with a fixed keep ratio, the adaptive keep ratio allows TGTP to adjust the pruning strength across different frames. Since target motion, scale, and background conditions may vary over time, different frames may require different numbers of retained tokens. This design makes the pruning process more flexible and better suited to the frame-wise nature of visual tracking.

Given the predicted motion offset, the target center in the current frame is estimated as

$$\hat{\mathbf{c}}_t = \mathbf{c}_{t-1} + (\Delta\hat{x}_t, \Delta\hat{y}_t).$$

For the i-th search token at pruning layer ℓ, let its spatial center be

$$\mathbf{p}_i^{\ell} = (u_i^{\ell}, v_i^{\ell}).$$

Based on the predicted target center and motion uncertainty, the motion prior for this token is computed as

$$\mathbf{P}_{t,i}^{\ell} = \exp\left(-\frac{(u_i^{\ell} - \hat{x}_t)^2}{2\hat{\sigma}_{x,t}^2} - \frac{(v_i^{\ell} - \hat{y}_t)^2}{2\hat{\sigma}_{y,t}^2}\right).$$

The layer-wise motion prior map is then denoted as

$$\mathbf{P}_t^{\ell} = \{\mathbf{P}_{t,i}^{\ell}\}_{i=1}^{N_{t,\ell}^{x}}.$$

This prior assigns larger scores to tokens near the predicted target center and smaller scores to distant background tokens. Compared with relying only on current-frame attention responses, the motion prior provides a temporal spatial bias derived from historical trajectories.

TGRL evaluates the sampled action using a reward function that considers tracking accuracy, computational cost, motion uncertainty, and target-token preservation:

$$\mathcal{R}_t = \lambda_{\mathrm{iou}}\mathrm{IoU}_t - \lambda_{\mathrm{flops}}\mathrm{FLOPs}_t - \lambda_{\sigma}\Omega_{\sigma,t} + \lambda_{\mathrm{keep}}\rho_t^{\mathrm{keep}}.$$

Here, IoU_t encourages accurate localization, FLOPs_t penalizes computational cost, and $\Omega_{\sigma,t}$ regularizes the motion uncertainty:

$$\Omega_{\sigma,t} = \hat{\sigma}_{x,t}^2 + \hat{\sigma}_{y,t}^2.$$

This term prevents the motion prior from becoming overly dispersed. Without this regularization, the policy may increase the uncertainty to assign high prior scores to too many tokens, weakening the selectivity of pruning.

The target-token preservation term is defined as

$$\rho_t^{\mathrm{keep}} = \frac{1}{|L_p|}\sum_{\ell \in L_p}\frac{|\mathcal{M}_t^{\ell} \cap \mathcal{G}_t^{\ell}|}{|\mathcal{G}_t^{\ell}|}.$$

Here, $\mathcal{M}_t^{\ell}$ denotes the retained token set at layer ℓ, and $\mathcal{G}_t^{\ell}$ denotes the target-related token set obtained by projecting the ground-truth box onto the token grid:

$$\mathcal{G}_t^{\ell} = \{i \mid \mathbf{p}_i^{\ell} \in \mathbf{b}_t^{\mathrm{gt}}\}.$$

If no token center falls inside the ground-truth box, the token closest to the ground-truth target center is included in $\mathcal{G}_t^\ell$ to avoid an empty target-token set. This reward component explicitly discourages the policy from pruning target-region tokens, improving the reliability of token selection.

We optimize TGRL with an actor-critic objective. The value branch estimates the expected return of the current state $V(\mathcal{S}_t)$. The advantage is computed as

$$\mathcal{D}_t = \mathcal{R}_t - V(\mathcal{S}_t).$$

The policy loss is defined as

$$\mathcal{L}_{\text{policy}} = -\left[\log \pi_\theta^m(\boldsymbol{\eta}_t|\mathcal{S}_t) + \log \pi_\theta^\gamma(k_t|\mathcal{S}_t)\right]\mathcal{D}_t.$$

When $\mathcal{D}_t > 0$, the sampled action performs better than expected and its probability is increased; otherwise, its probability is reduced. The value loss is

$$\mathcal{L}_{\text{value}} = \left(V(\mathcal{S}_t) - \mathcal{R}_t\right)^2.$$

To encourage exploration and prevent premature collapse to a single action, we further add entropy regularization. The final reinforcement learning loss is

$$\mathcal{L}_{\text{RL}} = \mathcal{L}_{\text{policy}} + \lambda_v \mathcal{L}_{\text{value}} - \lambda_e \mathcal{E}(\pi_\theta),$$

where λ_v and λ_e control the value loss and entropy regularization, respectively.

During inference, TGRL does not compute rewards or update the policy network. It only performs a forward pass using the historical trajectory state and deterministically predicts the motion prior and keep ratio. TGTP then fuses the motion prior with the current-layer template-search attention response to obtain the final token importance scores and performs token pruning at selected layers. This inference design keeps TGTP efficient and stable while reducing the number of search tokens and model FLOPs.

A.2 Additional Implementation Details

To ensure fair comparisons across different baseline trackers, TGTP is integrated as a lightweight plug-in module without changing the original tracking architecture, including the backbone, tracking head, template/search image resolution, region factors, and data-processing pipeline. TGTP is only inserted into several intermediate backbone layers to perform adaptive pruning on search tokens.

For different backbones, we use different pruning layers according to their network depths and block structures. The detailed settings are listed in Tab. 6.

Table 6: Pruning layers for different backbone networks.

Backbone	Total layers	Pruning layers
ViT-Tiny Wang et al. (2023a)	12	3, 6, 9
ViT-Base Dosovitskiy et al. (2021)	12	3, 6, 9
Fast-iTPN-T Tian et al. (2023)	14	7, 10, 13
Fast-iTPN-B Tian et al. (2023)	30	14, 20, 26
DeiT-Tiny Touvron et al. (2021)	12	3, 6, 9
HiViT-Base Zhang et al. (2023)	30	15, 20, 25

In the TGRL module, the trajectory state is constructed from recent target centers, box sizes, center displacements, and scale variations. The hidden dimension of the Trajectory Policy Network is set to 128. The maximum motion offset is set to $\delta_{\max} = 0.25$. The range of the Gaussian motion prior is set to $\sigma_{\min} = 0.08$ and $\sigma_{\max} = 0.35$, and the initial standard deviation of the continuous motion policy is set to 0.35. The initial keep ratio is set to 0.75. For the reward function, the weights of the IoU reward, FLOPs penalty, sigma regularization, and target-token keeping reward are set to 1.0, 0.05, 0.02, and 0.2, respectively. For the actor-critic objective, the weights of the value loss and entropy regularization are set to 0.2 and 0.01, respectively, and the overall loss weight of the RL branch is set to 0.5.

A.3 Additional Comparison Across Different Backbones

To further verify the adaptability of TGTP to different visual backbones, we provide additional performance comparisons in Tab. 7. This table covers trackers with different backbone networks,

Table 7: Additional performance comparison after integrating TGTP into trackers with different visual backbones.

Backbone	Method	Source	GOT-10k* AO	GOT-10k* SR50	GOT-10k* SR75	LaSOT AUC	LaSOT PN	LaSOT P	LaSOT_{ext} AUC	NFS30 AUC	FLOPs (G)↓	Params (M)↓	Speed (fps) GPU	Speed (fps) CPU
ViT-Tiny	FERMT-256	ECCV24	69.6	80.1	63.2	65.1	74.6	69.1	46.1	65.1	2.39G	7.98M	181fps	32fps
	+TGTP	**Ours**	**71.3**	**81.9**	**66.8**	**66.7**	**76.6**	**71.4**	**46.6**	**65.4**	2.02G	7.98M	199fps	40fps
	Δ		*1.7*↑	*1.8*↑	*3.6*↑	*1.6*↑	*2.0*↑	*2.3*↑	*0.5*↑	*0.3*↑	*15.4%*↓	*0%*↑	18fps↑	8fps↑
ViT-Base	OSTrack-256	ECCV22	74.9	84.2	73.0	67.9	76.9	73.3	47.6	-	29.1G	92.12M	133fps	8fps
	+TGTP	**Ours**	**75.0**	**85.1**	**73.4**	**68.6**	**78.1**	**74.5**	**47.0**	-	22.5G	92.14M	141fps	9fps
	Δ		*0.1*↑	*0.9*↑	*0.4*↑	*0.7*↑	*1.2*↑	*1.2*↑	*0.6* ↓	-	*22.7%*↓	*0.02%*↑	8fps↑	1fps↑
Fast-iTPN-T	MCITrack-T224	AAAI25	74.0	83.9	72.1	71.7	81.5	78.2	51.6	-	12.6G	32.44M	58fps	16fps
	+TGTP	**Ours**	**75.7**	**86.4**	**73.4**	**71.8**	**81.7**	**78.6**	**51.9**	-	11.3G	32.46M	63fps	19fps
	Δ		*1.7*↑	*2.5*↑	*1.3*↑	*0.1*↑	*0.2*↑	*0.4*↑	*0.3*↑	-	*10.3%*↓	*0.06%*↑	5fps↑	3fps↑
Fast-iTPN-B	MCITrack-B224	AAAI25	77.9	88.2	76.8	75.3	85.6	83.3	54.6	70.6	38.5G	88.03M	30fps	7fps
	+TGTP	**Ours**	**80.6**	**90.9**	**80.6**	**75.1**	**85.4**	**82.9**	**54.7**	**70.7**	34.5G	88.05M	34fps	8fps
	Δ		*2.7*↑	*2.7*↑	*3.8*↑	*0.2* ↓	*0.2* ↓	*0.4* ↓	*0.1*↑	*0.1*↑	*10.4%*↓	*0.02%*↑	4fps↑	1fps↑
HiViT-Base	EVPTrack-224	AAAI24	73.3	83.6	70.7	70.4	80.9	77.2	48.7	-	21.7G	73.65M	66fps	16fps
	+TGTP	**Ours**	**74.1**	**84.3**	**71.9**	**71.1**	**81.6**	**78.2**	**49.4**	-	18.2G	73.72M	73fps	18fps
	Δ		*0.8*↑	*0.7*↑	*1.2*↑	*0.7*↑	*0.7*↑	*1.0*↑	*0.7*↑	-	*16.1%*↓	*0.1%*↑	7fps↑	2fps↑

Backbone	Method	Source	UAVDT AUC	UAVDT P	DTB70 AUC	DTB70 P	UAV123 AUC	UAV123 P	Avg.Perf AUC	Avg.Perf P	FLOPs (G)↓	Params (M)↓	Speed (fps) GPU	Speed (fps) CPU
DeiT-Tiny	ORTrack-256	CVPR25	60.1	83.4	66.4	86.2	66.4	84.3	64.3	84.6	2.39G	7.98M	226fps	55fps
	+TGTP	**Ours**	**62.4**	**85.5**	**66.8**	**86.4**	**67.2**	**85.3**	**65.5**	**85.7**	1.91G	7.98M	238fps	61fps
	Δ		*2.3*↑	*2.1*↑	*0.4*↑	*0.2*↑	*0.8*↑	*1.0*↑	*1.2*↑	*1.1*↑	*20.1%*↓	*0%*↑	12fps↑	6fps↑

including ViT-Tiny Wang et al. (2023a), ViT-Base Dosovitskiy et al. (2021), DeiT-Tiny Touvron et al. (2021), Fast-iTPN-T Tian et al. (2023), Fast-iTPN-B Tian et al. (2023), and HiViT-Base Zhang et al. (2023), and evaluates them in terms of tracking performance, FLOPs, parameter number, and inference speed. Compared with the main experiments, this supplementary comparison further demonstrates the compatibility of TGTP with different backbone architectures.

As shown in Tab. 7, TGTP effectively reduces computational cost across different backbones while introducing only negligible additional parameters. For lightweight backbones, integrating TGTP into FERMT-256 Zheng et al. (2024) reduces FLOPs from 2.39G to 2.02G, keeps the parameter number at 7.98M, and improves GPU/CPU speed from 181fps/32fps to 199fps/40fps. When integrated into ORTrack-256 Wu et al. (2025), TGTP reduces FLOPs from 2.39G to 1.91G, keeps the parameter number at 7.98M, and improves GPU/CPU speed from 226fps/55fps to 238fps/61fps. These results show that TGTP can reduce FLOPs without introducing noticeable parameter overhead, while also improving practical inference speed on lightweight backbones.

For larger backbones, TGTP also maintains a small parameter overhead. On OSTrack-256 Ye et al. (2022), the parameter number only increases from 92.12M to 92.14M, while FLOPs are reduced from 29.1G to 22.5G and GPU/CPU speed is improved from 133fps/8fps to 141fps/9fps. On EVPTrack-224 Shi et al. (2024), the parameter number slightly increases from 73.65M to 73.72M, while FLOPs are reduced from 21.7G to 18.2G and GPU/CPU speed is improved from 66fps/16fps to 73fps/18fps. These results indicate that TGTP remains lightweight and can bring stable computational benefits even when applied to larger backbone networks.

In addition, TGTP also works effectively on Fast-iTPN backbones, whose structures are different from standard ViT backbones. On MCITrack-T224 Kang et al. (2025a) and MCITrack-B224 Kang et al. (2025a), TGTP reduces FLOPs by 10.3% and 10.4%, respectively, with only about 0.02M additional parameters. Meanwhile, both GPU and CPU inference speeds are improved. These results demonstrate that TGTP does not rely on a specific Transformer backbone. Instead, it can select suitable pruning layers according to the hierarchical structure of different backbones and serve as a general token pruning module for various visual tracking frameworks.

A.4 Attribute-Based Analysis on LaSOT

In addition to the overall performance comparison, we further analyze TGTP under different challenging attributes on LaSOT Fan et al. (2019), as shown in Fig. 4. The figure reports attribute-level AUC scores under both real-time RGB tracking and non-real-time RGB tracking settings, covering typical tracking challenges such as occlusion, deformation, motion blur, scale variation, background clutter, fast motion, low resolution, and viewpoint change.

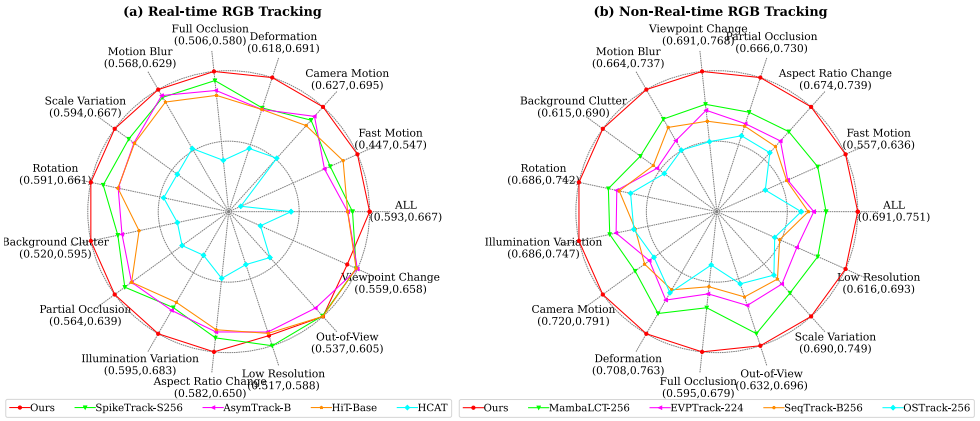


Figure 4: AUC scores of different attributes for real-time RGB and non-real-time RGB tracking.

For real-time RGB tracking, TGTP achieves higher or competitive AUC scores on most challenging attributes, especially under full occlusion, background clutter, rotation, illumination variation, and aspect ratio change. This indicates that TGTP does not merely reduce redundant tokens in simple scenarios, but can also preserve target-related tokens under complex interference, thereby maintaining reliable target representations.

For non-real-time RGB tracking, TGTP also shows strong robustness across different challenging attributes. Under difficult cases such as deformation, out-of-view, scale variation, low resolution, and background clutter, TGTP still maintains high AUC scores. This further demonstrates that trajectory-guided token pruning is not limited to a specific type of scene, but can use historical target states to provide effective priors for token selection under diverse tracking challenges.

Overall, Fig. 4 shows that TGTP reduces computational cost without weakening the robustness of trackers in complex scenarios. With the motion prior derived from historical trajectories, TGTP can more stably preserve target-related regions across different attributes, leading to improved or maintained tracking performance.

Table 8: Ablation studies on the number of historical frames, score fusion weights, candidate keep ratio sets, and reward components.

K	UAVDT AUC	UAVDT P
1	61.0	84.1
2	61.6	84.7
3	**62.4**	**85.5**
4	61.6	85.3

α	β	UAVDT AUC	UAVDT P
1.0	0.5	**62.4**	**85.5**
0.5	1.0	61.2	84.9
1.0	1.0	61.6	85.3
1.0	0.2	61.1	84.6

Candidate keep ratio set	UAVDT AUC	UAVDT P
{0.65,0.7,0.75}	61.8	85.1
{0.7,0.75,0.8}	**62.4**	**85.5**
{0.7,0.8,0.9}	61.1	84.5
{0.6,0.7,0.8,0.9}	60.7	83.8

Reward component	UAVDT AUC	UAVDT P
IoU reward	60.9	84.5
+ FLOPs penalty	61.4	84.7
+ Sigma regularization	61.7	85.2
+ Target-token reward	**62.4**	**85.5**

A.5 Detailed Ablation Studies

To provide a more detailed analysis of TGTP, we expand the ablation studies in Tab. 8. Different from the main paper, where the four ablation studies are summarized in one paragraph due to space limitation, we discuss each design separately in this section.

Number of Historical Frames. We first analyze the influence of the number of historical frames K. The historical trajectory is the basis for state construction and motion-prior prediction in TGRL, so K determines the temporal range observed by the policy network. When $K = 1$, the model only uses the target state from the previous frame and obtains 61.0 AUC and 84.1 P. As more historical frames are introduced, the model captures richer short-term motion trends and the performance improves. The best result is achieved when $K = 3$, reaching 62.4 AUC and 85.5 P. When K further increases to 4, the performance drops to 61.6 AUC, indicating that overly long historical trajectories may introduce outdated or noisy motion information. Therefore, we adopt $K = 3$ in the final setting to balance temporal information and motion-prior stability.

Score Fusion Weights. We further study the fusion weights between the attention response and the trajectory prior. Here, α controls the contribution of the current-layer template-search attention

response, while β controls the contribution of the trajectory-guided motion prior. The setting $\alpha = 1.0, \beta = 0.5$ achieves the best performance, with 62.4 AUC and 85.5 P. This suggests that current-frame attention remains the primary cue for token importance estimation, while the trajectory prior provides effective temporal guidance. When β is too large or α is too small, the model may over-rely on historical motion estimation and become sensitive to motion-prior errors. In contrast, when the trajectory prior is under-weighted, temporal information cannot be fully exploited. Therefore, a moderate trajectory-prior weight leads to a better balance between current-frame responses and historical motion cues.

Candidate Keep-ratio Set. We also compare different candidate keep-ratio sets for adaptive token pruning. The candidate set determines the range of pruning strengths available to the policy network. The set $\{0.7, 0.75, 0.8\}$ achieves the best result, showing that a moderate pruning range is more suitable for TGTP. When using $\{0.65, 0.7, 0.75\}$, the pruning strategy becomes more aggressive and may increase the risk of removing target-related tokens. When using $\{0.7, 0.8, 0.9\}$, larger keep ratios weaken the pruning effect and may reduce the benefit of adaptive selection. For the broader set $\{0.6, 0.7, 0.8, 0.9\}$, although the action space becomes larger, policy learning also becomes more difficult, leading to lower performance. These results indicate that a compact and proper keep-ratio set is important for stable policy learning and for balancing computational cost and tracking reliability.

Reward Components. Finally, we analyze the contribution of different reward components. With only the IoU reward, the model obtains 60.9 AUC and 84.5 P, suggesting that tracking overlap alone is insufficient to fully guide token pruning. Adding the FLOPs penalty improves AUC to 61.4, indicating that explicitly penalizing computational cost helps the policy learn more efficient pruning actions. After adding sigma regularization, AUC further improves to 61.7, showing that constraining motion uncertainty prevents the motion prior from becoming overly dispersed. With the target-token reward, the model achieves the best result, with 62.4 AUC and 85.5 P. This confirms the importance of explicitly encouraging the preservation of target-related tokens. Overall, these reward components guide policy learning from four aspects: tracking accuracy, computational cost, motion-prior stability, and target-token preservation.

Table 9: Ablation comparison between fixed and adaptive keep ratios.

#	Keep ratio	UAVDT	
		AUC	P
1	Fixed 0.7	60.9	83.6
2	Fixed 0.75	61.6	84.7
3	Fixed 0.8	61.3	84.6
4	Adaptive	**62.4**	**85.5**

Fixed vs. Adaptive Keep Ratio. We further compare fixed keep ratios with the adaptive keep ratio learned by TGRL. A fixed keep ratio uses the same token retention ratio for all frames, while the adaptive keep ratio is dynamically selected according to the historical trajectory state. As shown in Tab. 9, the adaptive strategy achieves the best performance, with 62.4 AUC and 85.5 P, outperforming fixed keep ratios of 0.7, 0.75, and 0.8. Compared with the fixed keep ratio of 0.75, the adaptive strategy brings improvements of 0.8 AUC and 0.8 P. This indicates that different frames may require different token retention ratios due to variations in target motion, scale, and background complexity. By selecting the keep ratio in a frame-wise adaptive manner, TGTP can better balance token reduction and tracking reliability, leading to improved tracking performance.

A.6 Visualization of Tracking Results

To intuitively show the localization performance of TGTP during tracking, we provide bounding-box visualization results under different modalities in Fig. 5. The figure includes RGB, RGB-T, RGB-D, and RGB-E tracking examples, where the red boxes denote our method, the green and blue boxes denote other SOTA methods, and the yellow boxes denote the ground truth.

It can be observed that our method predicts bounding boxes closer to the ground truth in various challenging scenarios. For example, under target appearance variation, background interference, modality noise, or target scale changes, other SOTA methods may suffer from localization shifts or inaccurate scale estimation, while our method still maintains stable target localization. This indicates that TGTP does not destroy key target representations when performing token pruning.

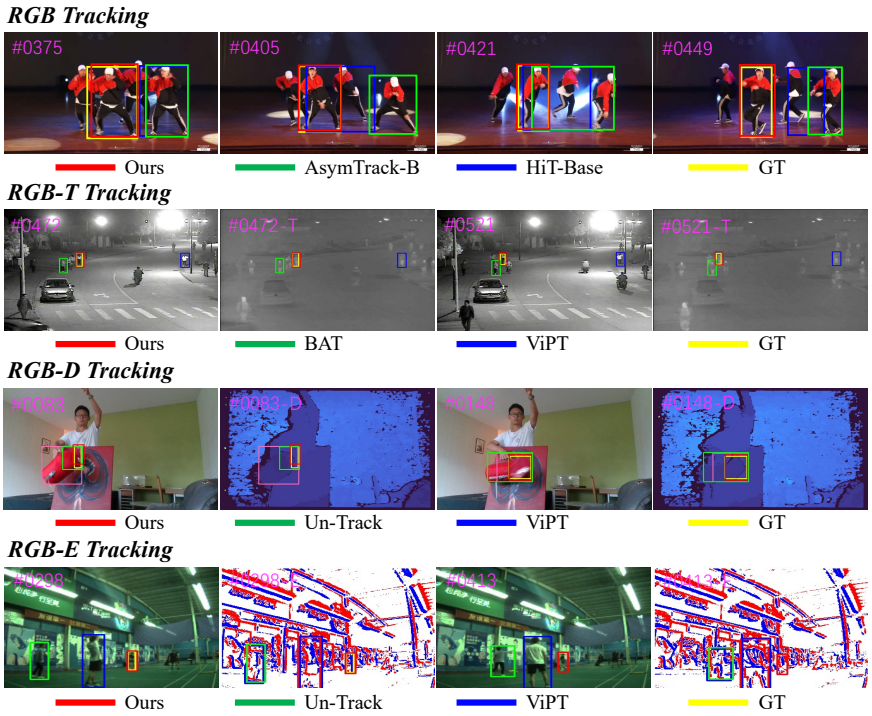


Figure 5: Visualization of tracking results under RGB and RGB-T/D/E tracking scenarios.

This stability mainly comes from the trajectory-guided token selection mechanism. TGTP uses historical target trajectories to generate motion priors, making the tracker focus more on regions where the target is likely to appear in the current frame and reducing the interference from background tokens. Even under different modalities such as thermal infrared, depth, and event streams, trajectory information can still provide cross-frame spatial constraints and help preserve target-related tokens. Therefore, Fig. 5 further shows from a visualization perspective that TGTP not only reduces computational cost, but also maintains reliable localization ability under different modalities and complex scenarios. This is consistent with the quantitative results and further verifies the effectiveness of TGTP as a general token pruning module.

A.7 Limitation Analysis

Although TGTP shows good generality and effectiveness across different visual tracking scenarios, it still has certain limitations. One main limitation lies in the selection of the token keep ratio. In the current design, TGRL formulates the keep ratio as a discrete action and selects it from a predefined candidate set. This design avoids the instability of continuous ratio prediction and makes the computational cost under different pruning strengths easier to control. However, the discrete candidate set also limits the expressiveness of the policy, since the model can only choose from a limited number of pruning strengths.

Specifically, when target motion, scale variation, or background complexity changes smoothly across frames, a fixed candidate set may not always provide the optimal token keep ratio. For some frames, the model may only need a slight adjustment of the keep ratio, but the current discrete policy can only select the closest candidate value. Although this design is beneficial for training stability and experimental controllability, it may limit the ability of TGTP to further optimize the trade-off between computational cost and tracking accuracy in more complex scenarios.

In future work, more flexible keep-ratio modeling strategies can be explored. For example, the keep ratio can be formulated as a continuous action and directly predicted by the policy network. Another possible direction is to design hierarchical or dynamic candidate sets, allowing the model to adaptively adjust the available pruning range according to scene complexity. In addition, hardware-aware constraints, such as real inference latency or energy consumption, can be incorporated into the reward design, making the token pruning policy more suitable for real-world deployment.