PURETRACK: TEMPLATE PURIFICATION WITH TEMPORAL LEARNING FOR EFFICIENT TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Efficient tracking requires a careful balance between inference speed and tracking accuracy. Many efficient trackers adopt a single fixed template to reduce computational cost. However, background and distractor content in the template can become entangled with the target representation and induce false responses during subsequent template-search interactions. In addition, a fixed template cannot effectively adapt to appearance changes over time. To address these issues, we propose **PureTrack**, an efficient framework that combines template purification with temporal learning to improve template reliability and temporal adaptability. First, PureTrack decouples template initialization from subsequent template-search interaction and establishes an independent template-processing stage. An Attention-Guided Template Purification (AGTP) module then uses global template-relevance cues from the spatio-temporal token to suppress background interference while preserving the complete spatial structure of the template, producing more reliable response maps. Second, the spatio-temporal token is propagated across frames as a compact state to supplement the temporal context missing from the fixed template. Finally, building on the cleaner response maps produced by AGTP, we introduce a Bidirectional Temporal Consistency (BTC) learning paradigm to enforce cross-frame localization consistency. BTC spatially aligns response maps according to target displacement and imposes bidirectional consistency constraints during training, improving localization stability under target motion without introducing additional inference-time computation. Experiments show that PureTrack improves the reliability and temporal stability of fixed-template representations while achieving a favorable accuracy-efficiency trade-off across different platforms. It achieves 72.5% AO on GOT-10k and 208/76/84 FPS on GPU/CPU/AGX.

1 INTRODUCTION

Visual object tracking aims to continuously estimate the target state in subsequent frames given its initial location in the first frame. Recent high-accuracy trackers usually rely on complex architectures and dense feature interactions, which introduce considerable computation and memory costs, limiting their deployment in resource-constrained scenarios. Therefore, many efficient trackers adopt a fixed-template paradigm to reduce computational overhead for high-speed inference. Although this strategy significantly improves inference efficiency, target representation is inevitably constrained by the quality and temporal rigidity of the fixed template.

The initial template is typically cropped from a region surrounding the target and therefore inevitably contains background content, similar objects, and other distractors. Once encoded, these irrelevant contents become entangled with the target representation and may induce false responses during online tracking. Existing studies have attempted to improve template quality through template selection, target-background modeling, and template representation compression to alleviate background contamination Cui et al. (2022); Gao et al. (2022); Zhang et al. (2024); Liu et al. (2024); Cai et al. (2023); Zhang et al. (2025); Wu et al. (2026). However, for efficient tracking, most trackers adopt a single fixed template, which prevents them from mitigating template contamination via online updating or multi-template interaction and may bring extra storage and computation costs. Moreover, directly cropping, removing, or compressing template tokens can reduce background interference but may discard spatial cues useful for target localization, such as object boundaries, local structures, and spatial positions. Therefore, how to effectively purify a single initial fixed template prior to the initial

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.

template-search interaction, while retaining its full spatial structure, remains a critical challenge. To address this issue, PureTrack performs one-time purification of the fixed template, suppressing background interference while preserving its full spatial structure.
Although a fixed template reduces the computational cost of online tracking, it only preserves the static appearance of the first frame and therefore struggles to adapt to target variations in subsequent frames. Thus, how to introduce cross-frame temporal information while maintaining stability of the fixed-template representation remains a challenge for tracking.

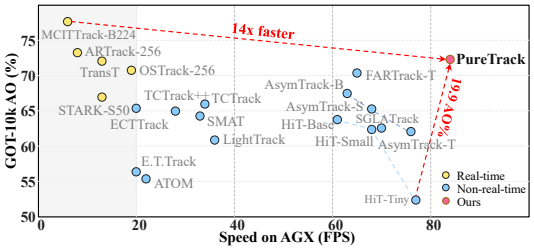


Figure 1: Comparison of our PureTrack with real-time and non-real-time trackers in terms of the accuracy-efficiency trade-off on NVIDIA Jetson AGX Xavier.

Existing methods typically adapt to target appearance changes by updating the template, maintaining historical templates, or constructing temporal memories Zhang et al. (2019); Mayer et al. (2021); Park et al. (2025); Fu et al. (2021).However, these methods usually require additional template storage and computation. Moreover, they update historical representations based on tracking predictions, which causes errors to accumulate over time and compromise their reliability. Other approaches do not directly update the fixed template, but instead introduces dynamic information through temporal feature propagation or compact temporal states Cao et al. (2022); Zheng et al. (2024); Xue et al. (2026). Compared with direct template updating, these methods enhance temporal adaptability with lower computational overhead while preserving the stability of the fixed template. Inspired by these methods, PureTrack introduces a lightweight spatio-temporal token to aggregate and propagate localization-relevant spatial and temporal information across frames, thereby supplementing the temporal information missing from the fixed template.

Compact spatio-temporal states can supplement fixed templates with dynamic information. Even so, further constraints are required to ensure that they consistently transmit meaningful temporal cues for target localization. To enhance the modeling of cross-frame localization relationships, existing studies have applied consistency learning at different levels, including target trajectories, bounding boxes, and response maps Shen et al. (2022); Zhang et al. (2023). However, trajectory and bounding-box-level consistency constraints typically operate on final predictions and therefore cannot directly optimize the underlying spatial localization relationships encoded in response maps. Methods that rely on network predictions to align response maps may also introduce additional online computation and propagate alignment errors. Therefore, during training, we use the ground-truth target displacement to spatially align the relatively cleaner response maps and directly enforce cross-frame response consistency without introducing any additional inference-time computational overhead.

To address the above spatial and temporal challenges, we propose **PureTrack**, an efficient framework integrating template purification with temporal learning for reliable fixed-template visual tracking. Specifically, PureTrack makes the following three contributions. First, we decouple template initialization from subsequent template-search interaction to establish an independent template-processing stage. Based on this decoupled architecture, we introduce the Attention-Guided Template Purification module (AGTP), which purifies the initial template representation using global template-relevance cues provided by the spatio-temporal token. This produces a more reliable and discriminative fixed-template representation, supporting more accurate response prediction. Second, the spatio-temporal token is further propagated across frames as a compact state to supplement the temporal information missing from the fixed template. Third, although the purified template improves the reliability of individual responses, temporal state propagation alone does not explicitly constrain localization consistency for moving targets across consecutive frames. We therefore introduce Bidirectional Temporal Consistency (BTC) learning during training to encourage the propagated temporal information to remain focused on target localization and to facilitate stable cross-frame localization of moving targets. Specifically, the proposed mechanism aligns adjacent response maps according to the ground-truth target displacement and imposes bidirectional consistency constraints, without introducing additional inference-time computational overhead. Experiments show that PureTrack achieves a favorable trade-off between tracking performance and inference speed. As shown in Figure 1, it achieves 72.5% AO on GOT-10k and runs at 84 FPS on the NVIDIA Jetson AGX Xavier.

2 Related Work

Efficient Visual Tracking. Practical tracking deployment demands both high localization accuracy and low latency. Early Siamese trackers achieve high inference efficiency but exhibit limited representation modeling capability Bertinetto et al. (2016); Li et al. (2018). Transformer-based single-stream trackers significantly boost tracking robustness yet sacrifice deployment efficiency Ye et al. (2022); Cui et al. (2022); Yan et al. (2021a). Existing efficient tracking methods optimize the accuracy-speed trade-off from network architecture, interaction paradigms and inference strategies Borsuk et al. (2022); Blatter et al. (2023); Wei et al. (2024); Yan et al. (2021b); Kang et al. (2023); Cui et al. (2023); Zhu et al. (2025); Li et al. (2023); Wu et al. (2024); Xue et al. (2025); Wang et al. (2026). However, most computation-oriented optimization schemes assume reliable fixed-template representations while largely ignoring inherent template defects. Targeting this core representation quality limitation of fixed templates, this work proposes an efficient and robust tracking framework.

Template Representation and Temporal Modeling. Existing trackers improve template representations through online template selection Cui et al. (2022), target–background modeling Gao et al. (2022); Zhang et al. (2024); Yang et al. (2023), template filtering Liu et al. (2024), multi-template interaction Cai et al. (2023); Zhang et al. (2025), and historical template compression Wu et al. (2026). Aba-ViTrack Li et al. (2023) also reduces background interference through background-aware token computation. Unlike these approaches, PureTrack purifies a single initial template before template–search interaction, suppressing background noise while preserving its spatial structure.

Temporal information is commonly incorporated via template updating or historical memory Zhang et al. (2019); Mayer et al. (2021); Park et al. (2025); Fu et al. (2021), approaches that increase storage and computation overhead. Alternatively, temporal feature and token propagation Cao et al. (2022); Zheng et al. (2024); Xue et al. (2026) can supply dynamic context for the fixed template. The latter strategy avoids directly modifying the fixed template and offers a lightweight way to improve temporal adaptability. Building on this direction, PureTrack introduces a lightweight spatio-temporal token that jointly aggregates spatial structure and temporal dynamics, enabling localization-relevant information to be propagated across frames. Unlike purely temporal representations, this token incorporates spatial cues from the target representation, allowing it to complement the stable spatial prior from the fixed template.

Temporal Consistency Learning. Consistency learning promotes representation robustness by enforcing consistent outputs for semantically equivalent inputs, and has been widely used in self-supervised learning and video modeling Tarvainen & Valpola (2017); Bardes et al. (2022); Dwibedi et al. (2019). In visual tracking, ULAST Shen et al. (2022) enforces consistency on tracking trajectories and bounding boxes. Then Siam-STM Zhang et al. (2023) adopts a temporal consistency module. This module constrains abrupt changes in inter-frame response maps and exploits historical response context during inference, thus incurring additional inference-time computation. In contrast, we introduce the BTC learning that uses ground-truth target displacements to align response maps during training, without adding any inference-time computational cost.

3 Method

3.1 Overall Architecture

The overall framework is presented in Figure 2. Given the target template $Z \in \mathbb{R}^{H_z \times W_z \times 3}$ and search region $X \in \mathbb{R}^{H_x \times W_x \times 3}$, the tracker aims to estimate the object bounding box $B \in \mathbb{R}^4$ in X.

PureTrack decouples template initialization from template-search interaction. During training, an initial template and temporally correlated search regions are sampled from the same video. The template branch independently processes template together with the initial spatio-temporal token. AGTP then purifies the template representation using global template-relevance cues extracted from the final template-encoding layer, yielding the fixed spatial template memory Z_0^p, or each search region, the fusion module conducts feature interaction with the purified template memory Z_0^p, the current search feature, and the spatio-temporal token. The tracking head further outputs the corresponding response map as well as the target bounding box. The fixed template memory offers a stable spatial prior, whereas the spatio-temporal token propagates temporal context across search regions. In addition to the basic tracking loss, BTC imposes bidirectional consistency constraints

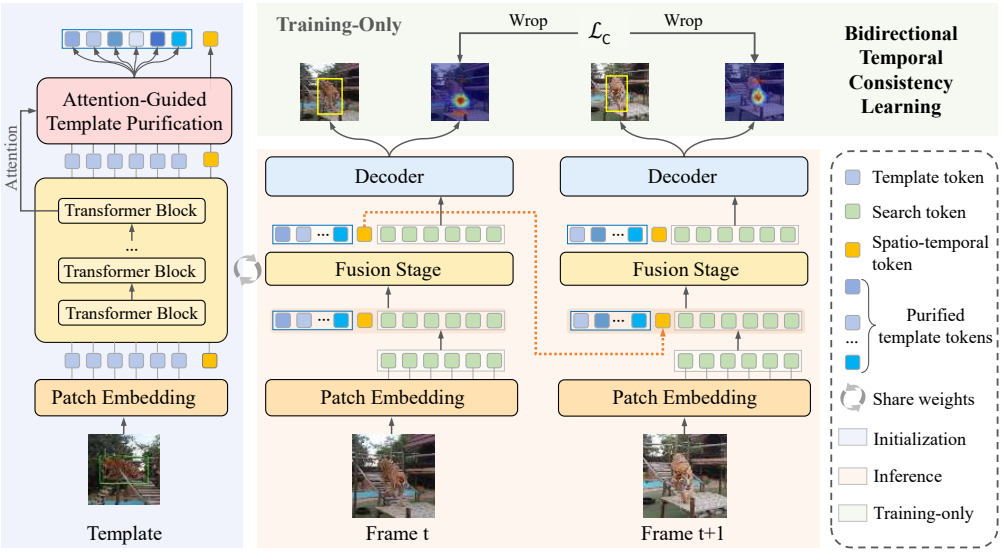


Figure 2: Overall architecture of PureTrack. It includes template initialization and online tracking. AGTP re-weights template tokens for purified reusable template memory. Spatio-temporal token is updated frame-by-frame in online tracking for localization temporal context. Bidirectional consistency loss is used only in training to enhance temporal representations and target localization.

on the corresponding response maps according to the ground-truth target displacement between temporally correlated search regions. During inference, the initial fixed template is encoded and purified only once, and Z_0^p is cached and reused throughout the entire video sequence. For the search region X_t of the t-th frame, PureTrack feeds it into the fusion module together with the template memory Z_0^p and the spatio-temporal token from the previous frame. The output search feature is fed into the tracking head for target localization, and the updated spatio-temporal token is stored for the next frame. BTC is enabled only during training and is not involved in inference.

3.2 Attention-Guided Template Purification

The initial template is cropped from the image patch covering the target together with surrounding areas. Even with accurate bounding-box annotation, it inevitably contains background and irrelevant content, which causes distractor responses in matching. To alleviate this problem, AGTP purifies the initial template before it interacts with search regions. Instead of discarding background information completely, our purification strategy performs soft modulation over template tokens. Given the initial template Z_0 and the initial spatio-temporal token C_0, the template branch jointly encodes them. And it then takes the spatio-temporal token as a global template-relevance query to extract its attention toward the template tokens:

$$[\tilde{Z}_0, \tilde{C}_0], A = \Phi_z([Z_0, C_0]), \qquad r = \mathcal{N}_\mu(A) \tag{1}$$

where $\Phi_z(\cdot)$ denotes the template-encoding transformer blocks, $[\cdot]$ represents token concatenation, $\tilde{Z}_0$ stands for the encoded template representation, A indicates the attention weights from the spatio-temporal token to template tokens, and $\mathcal{N}_\mu(\cdot)$ denotes mean normalization. With the above attention outputs, the normalized relevance scores r are converted into purification gains, which are used to reweight the template tokens and obtain the purified template memory:

$$Z_0^p = \tilde{Z}_0 \odot \psi(r), \qquad \psi(r) = \frac{1+r}{2} \tag{2}$$

where $\odot$ denotes token-wise multiplication with broadcasting along the channel dimension. AGTP enhances target-relevant template tokens and suppresses background-dominated ones without deleting or merging any tokens. It thus alleviates template contamination while preserving the full spatial structure and localization cues of the initial template. As the initial template remains fixed throughout tracking, Z_0^p is computed only once during initialization and cached for subsequent frames.

3.3 SPATIO-TEMPORAL TOKEN PROPAGATION

The AGTP-purified fixed template provides stable spatial prior knowledge but cannot model online target appearance variations. Inspired by temporal token propagation, PureTrack propagates a lightweight spatio-temporal token across frames. For the t-th frame, the search tokens X_t, Z_0^p, and C_{t-1} from the previous frame are fed into the subsequent Transformer fusion blocks:

$$[\tilde{X}_t, \tilde{Z}_0^p, C_t] = \Phi_f\left([X_t, Z_0^p, C_{t-1}]\right) \tag{3}$$

where $\Phi_f(\cdot)$ denotes the feature-fusion mapping implemented by the subsequent transformer blocks. The output $\tilde{X}_t$ represents the fused search features used for target localization, while C_t denotes the updated spatio-temporal token propagated to the next frame. Through interaction with the purified template memory and the current search tokens, C_t jointly aggregates the stable spatial prior of the target and its current temporal variations. The fixed template memory remains unchanged throughout tracking, whereas the spatio-temporal token is continuously updated to propagate localization-relevant information across frames. This design supplements the temporal information missing from the fixed template without online template updating or additional template storage.

3.4 BIDIRECTIONAL TEMPORAL CONSISTENCY LEARNING

The core idea of BTC learning is to encourage consistent target responses across temporally correlated frames despite appearance variations and target motion. However, the target may appear at different spatial locations in different frames. Directly comparing response maps in their original coordinate systems will introduce inaccurate supervision. To address this issue, we first align the response maps according to the ground-truth target displacement and then impose the consistency constraint. During training, temporally correlated search regions X_i are sampled from the same video. Their search features are fused with the purified template memory and the propagated spatio-temporal token. Let $\hat{X}_i$ denote the fused search feature corresponding to X_i. The tracking head $\mathcal{D}(\cdot)$ produces the response map M_i, which is further normalized as:

$$M_i = \mathcal{D}(\hat{X}_i), \qquad \bar{M}_i = \mathcal{N}_{\max}(M_i), \qquad i \in \{1, 2\} \tag{4}$$

where $\mathcal{N}_{\max}(\cdot)$ denotes per-map maximum normalization. We obtain the target centers c_1 and c_2 from the corresponding ground-truth bounding boxes and compute the normalized center displacement as Δc. The normalized response maps are then aligned in both directions:

$$\tilde{M}_{1\to 2} = \mathrm{Warp}(\bar{M}_1, \Delta c), \qquad \tilde{M}_{2\to 1} = \mathrm{Warp}(\bar{M}_2, -\Delta c) \tag{5}$$

where $\mathrm{Warp}(\cdot, \Delta c)$ denotes the spatial transformation that compensates for the target displacement. The bidirectional temporal consistency loss is defined as:

$$\mathcal{L}_{\mathrm{b}} = \frac{1}{2}\left[\left\|\tilde{M}_{1\to 2} - \bar{M}_2\right\|_1 + \left\|\tilde{M}_{2\to 1} - \bar{M}_1\right\|_1\right] \tag{6}$$

The bidirectional formulation provides symmetric supervision for the two temporally correlated samples and exploits their correspondence in both directions. By aligning the response maps before comparison, BTC Learning encourages stable target localization under inter-frame motion. Since BTC learning is applied only during training, no response-map alignment or additional computational branch is required during inference.

3.5 TRAINING LOSS

PureTrack adopts a center-based prediction head for localization. Taking fused search feature, the head predicts center response map, size-regression map and offset-regression map. The peak position on center response map gives target center, and bounding-box is decoded from corresponding size and offset outputs. The total training loss combines basic tracking loss and BTC loss:

$$\mathcal{L} = \sum_{i=1}^{2}\left(\lambda_{\mathrm{focal}}\mathcal{L}_{\mathrm{focal}}^{(i)} + \lambda_1\mathcal{L}_1^{(i)} + \lambda_{\mathrm{giou}}\mathcal{L}_{\mathrm{giou}}^{(i)}\right) + \lambda_{\mathrm{b}}\mathcal{L}_{\mathrm{b}} \tag{7}$$

where $\mathcal{L}_{\mathrm{focal}}^{(i)}$ supervises center response map, $\mathcal{L}_1^{(i)}$ and $\mathcal{L}_{\mathrm{giou}}^{(i)}$ constrain coordinate error and overlap between predicted box and ground-truth. λ_{b} weights consistency term $\mathcal{L}_{\mathrm{b}}$, which acts on center response maps of paired search regions to enforce displacement-compensated cross-frame response consistency. The basic tracking loss is computed independently for each search region, while the BTC loss jointly regularizes their temporal correspondence.

4 EXPERIMENTS

PureTrack is implemented using Python 3.8 and PyTorch 1.11. Training is conducted on 2 Intel Xeon Gold 6354 @3.00 GHz CPUs and 4 NVIDIA RTX 4090 GPUs. Inference speed was evaluated on three platforms: a single GPU, a CPU, and the Jetson AGX Xavier.

4.1 IMPLEMENTATION DETAILS

Model. PureTrack is built upon the 8-layer Fast-iTPN-Tiny backbone with a hidden dimension of 384, including four Transformer layers for high-level feature modeling and template–search interaction. It adopts a center-based prediction head for target localization. The template and search regions are resized to 112×112 and 224×224, respectively. A learnable 384-dimensional spatio-temporal token is introduced to aggregate target information shared by temporally ordered search frames. During data preprocessing, the template and search regions are cropped using spatial scaling factors of 2.0 and 4.0, respectively.

Training. We use five tracking datasets for training, including LaSOT Fan et al. (2019), COCO2017 Lin et al. (2014), TrackingNet Muller et al. (2018), GOT-10k Huang et al. (2019), and VASTTrack Peng et al. (2024). The model takes one template and two search regions as inputs, and the model is trained for 180 epochs. In each epoch, 100,000 image pairs are randomly sampled for training with a batch size of 64. We employ AdamW as the optimizer with an initial learning rate of 2×10^{-4} and a weight decay of 1×10^{-4}. The two search regions are leveraged to establish forward and backward temporal correspondences and calculate our proposed BTC loss. The weights of the focal, ℓ_1, and GIoU losses follow SUTrack Chen et al. (2025), while the BTC loss weight is set to $\lambda_{\mathrm{b}} = 0.2$.

Inference. During inference, we apply a Hann window penalty to the response map for suppressing abrupt target displacement and improve temporal stability.

4.2 STATE-OF-THE-ART COMPARISONS

We report the inference speeds of all compared trackers on GPU, CPU and Jetson AGX Xavier platforms. Trackers are divided into real-time and non-real-time groups for comparison according to their AGX speed. PureTrack achieves the highest real-time performance score of all benchmarks.

GOT-10k. GOT-10k Huang et al. (2019) is a large-scale short-term tracking benchmark with over 10000 video sequences. Table 1 shows that PureTrack achieves the best AO among real-time trackers. It outperforms the second-ranked FARTrack-Tiny Wang et al. (2026) by 1.9% in AO and 4.9% in $\mathrm{SR}_{0.75}$. Although non-real-time trackers like MCITrack-B224 Kang et al. (2025) achieve higher AO, they run at merely 6 FPS on AGX. In comparison, PureTrack only has a 5.4% AO degradation while reaching 84 FPS on AGX, roughly 14× faster, and preserves competitive tracking accuracy.

LaSOT. LaSOT Fan et al. (2019) is a challenging long-term tracking benchmark consisting of 280 test sequences. As shown in Table 1, PureTrack attains 67.8% AUC, 77.5% P_{Norm} and 73.2% P, while maintaining the highest inference speed across GPU, CPU and AGX at 208 FPS, 76 FPS and 84 FPS, respectively. Although AsymTrack-T Zhu et al. (2025) achieves competitive speed, PureTrack surpasses it by 7.0% in AUC, 8.8% in P_{Norm} and 12.0% in P. Compared with Seqtrack Chen et al. (2023), PureTrack sacrifices only 2.1% AUC but delivers a 2.6× GPU speedup.

TrackingNet. TrackingNet Muller et al. (2018) is a large-scale short-term tracking benchmark with 511 test sequences. From Table 1, PureTrack obtains competitive results with 82.5% AUC, 87.2% P_{Norm} and 80.3% P. It outperforms FARTrack-Tiny Wang et al. (2026) by 1.8% in AUC and runs 1.3× faster on GPU. PureTrack is 1.7% lower in AUC than ARTrack Wei et al. (2023), the fastest non-real-time tracker. Nevertheless, it runs 4.4× faster on the AGX, achieving a better trade-off between accuracy and efficiency.

$\mathrm{LaSOT}_{\mathrm{ext}}$. $\mathrm{LaSOT}_{\mathrm{ext}}$ Fan et al. (2020) is a more challenging extension of LaSOT, consisting of complex scenarios. According to Table 2, PureTrack reaches a leading AUC of 48.4%. It outperforms FARTrack-Tiny Wang et al. (2026) by 3.4% and surpasses AsymTrack-B Zhu et al. (2025) by 3.8%. Since $\mathrm{LaSOT}_{\mathrm{ext}}$ includes categories disjoint from the original LaSOT training set, this gain verifies PureTrack's strong category-level generalization capability.

Table 1: State-of-the-art comparison on GOT-10k, LaSOT, and TrackingNet benchmarks. In each column, the best and second-best results are highlighted in **bold** and underlined, respectively.

	Method	GOT-10k			LaSOT			TrackingNet			PyTorch Speed (*fps*)		
		AO	$SR_{0.5}$	$SR_{0.75}$	AUC	P_{Norm}	P	AUC	P_{Norm}	P	GPU	CPU	AGX
Real-time	**PureTrack (Ours)**	**72.5**	**82.5**	**68.7**	**67.8**	**77.5**	**73.2**	**82.5**	**87.2**	**80.3**	**208**	**76**	**84**
	FARTrack-Tiny Wang et al. (2026)	70.6	81.0	63.8	63.2	71.6	66.7	80.7	85.6	77.5	165	68	65
	SGLATrack Xue et al. (2025)	65.2	75.2	59.0	62.8	71.9	31.0	79.6	84.9	76.6	177	73	70
	AsymTrack-B Zhu et al. (2025)	67.7	76.6	61.4	64.7	73.0	67.8	80.0	84.5	77.4	164	45	63
	AsymTrack-S Zhu et al. (2025)	65.5	74.8	58.9	62.8	71.2	64.8	77.9	82.2	74.0	169	66	68
	AsymTrack-T Zhu et al. (2025)	62.3	71.3	54.7	60.8	68.7	61.2	76.2	80.9	71.6	187	**76**	76
	ECTTrack Xu et al. (2025)	65.6	75.0	60.7	62.4	71.5	66.3	78.8	84.6	76.5	86	30	20
	SMAT Gopal & Amer (2024)	64.5	74.7	57.8	61.7	71.1	64.6	78.6	84.2	75.6	101	36	33
	LiteTrack-B4 Wei et al. (2024)	65.2	74.7	57.7	62.5	72.1	65.7	79.9	84.9	76.6	184	33	51
	HiT-Base Kang et al. (2023)	64.0	72.1	58.1	64.6	73.3	68.1	80.0	84.4	77.3	132	35	61
	HiT-Small Kang et al. (2023)	62.6	71.2	54.4	60.5	68.3	61.5	77.7	81.9	73.1	138	46	68
	HiT-Tiny Kang et al. (2023)	52.6	59.3	42.7	54.8	60.5	52.9	74.6	78.1	68.8	151	53	77
	E.T.Track Blatter et al. (2023)	56.6	64.6	42.5	59.1	66.8	60.1	72.5	77.8	69.5	54	38	20
	TCTrack Cao et al. (2022)	66.2	75.6	61.0	60.5	69.3	62.4	74.8	79.6	73.3	124	26	34
	TCTrack++ Cao et al. (2023)	65.2	75.2	59.0	43.5	48.4	41.4	-	-	-	131	27	28
	LightTrack Yan et al. (2021b)	61.1	71.0	54.3	53.8	60.5	53.7	72.5	77.8	69.5	83	50	36
	ATOM Danelljan et al. (2019)	55.6	63.4	40.2	51.5	57.6	50.5	70.3	77.1	64.8	119	13	22
Non-real-time	MCITrack-B224 Kang et al. (2025)	77.9	88.2	76.8	75.3	85.6	83.3	86.3	90.9	86.1	36	7	6
	SeqTrack-B256 Chen et al. (2023)	74.7	84.7	71.8	69.9	79.7	76.3	83.3	88.3	82.2	80	4	-
	ARTrack-256 Wei et al. (2023)	73.5	82.2	70.9	70.4	79.5	76.6	84.2	88.7	83.5	32	5	8
	OSTrack-256 Ye et al. (2022)	71.0	80.4	68.2	69.1	78.7	75.2	83.1	87.8	82.0	113	16	19
	STARK-S50 Yan et al. (2021a)	67.2	76.1	61.2	-	-	-	80.3	85.1	-	58	14	13
	TransT Chen et al. (2021)	72.3	82.4	68.2	64.9	73.8	69.0	81.4	86.7	80.3	82	6	13
	DiMP Bhat et al. (2019)	61.1	71.7	49.2	56.9	65.0	56.7	74.0	80.1	68.7	-	-	17
	SiamFC++ Xu et al. (2020)	59.5	69.5	47.9	54.5	-	54.7	75.4	80.0	70.5	-	-	-
	SiamRPN++ Li et al. (2019)	51.7	61.6	32.5	49.6	56.9	49.1	73.3	80.0	69.4	-	-	11

UAV123. UAV123 Mueller et al. (2016) is a typical UAV-captured tracking benchmark, characterized by frequent camera shake, scale variation, small targets and complex aerial backgrounds. As shown in Table 2, PureTrack achieves 68.9% AUC. It surpasses the second-best ECTTrack Xu et al. (2025), AsymTrack-B Zhu et al. (2025) and FARTrack-Tiny Wang et al. (2026) by 1.9%, 2.4% and 3.1%, respectively. These gains on aerial sequences verify PureTrack's robustness against small targets, viewpoint shifts, background clutter and intensive camera motion.

NFS. NFS Kiani Galoogahi et al. (2017) is a high-frame-rate benchmark for fast-motion tracking, featuring frequent motion blur and large inter-frame displacements. As shown in Table 2, PureTrack obtains 67.7% AUC.

Table 2: State-of-the-art comparison on more benchmarks.

Methods	$LaSOT_{ext}$	NFS	UAV123
ATOM Danelljan et al. (2019)	37.6	58.4	64.2
LightTrack Yan et al. (2021b)	-	55.3	62.5
FEAR Borsuk et al. (2022)	-	61.4	-
E.T. Track Blatter et al. (2023)	-	59.0	62.3
HiT-Tiny Kang et al. (2023)	35.8	53.2	58.7
HiT-Small Kang et al. (2023)	40.4	61.8	63.3
HiT-Base Kang et al. (2023)	44.1	63.6	65.6
MixformerV2 Cui et al. (2023)	43.6	-	65.8
LiteTrack-B4 Wei et al. (2024)	-	63.4	66.4
SMAT Gopal & Amer (2024)	-	62.0	64.3
ECTTrack Xu et al. (2025)	39.9	61.1	67.0
AsymTrack-T Zhu et al. (2025)	42.5	63.3	64.6
AsymTrack-S Zhu et al. (2025)	43.3	64.9	65.6
AsymTrack-B Zhu et al. (2025)	44.6	64.4	66.5
FARTrack-Tiny Wang et al. (2026)	45.0	66.9	65.8
PureTrack (Ours)	**48.4**	**67.7**	**68.9**

Specifically, it outperforms FARTrack-Tiny Wang et al. (2026), AsymTrack-S Zhu et al. (2025) and LiteTrack-B4 Wei et al. (2024) by 0.8%, 2.8% and 4.3%. This promising result shows that the temporal correspondences learned by PureTrack work well under fast-motion conditions.

4.3 Experimental Analyses

We further conduct experiments on GOT-10k Huang et al. (2019) to analyze the contribution of each core component to PureTrack. All results are reported in Table 3 where default settings of PureTrack are marked in gray.

Analysis of the decoupled architecture. The baseline model (#1) achieves 70.6% AO at 162 FPS. With the decoupled architecture (#2), AO rises to 71.2% and speed reaches 211 FPS, yielding 0.6% AO gain and 49 FPS improvement. Computational cost decreases from 2.82G (#1) to 1.17G (58.5% reduction), with inference speed improved by 30.2% (162 FPS to 211 FPS). This result validates that decoupling template initialization from template-search interaction and processing the template independently at initialization eliminates redundant online computation. Such a design jointly improves both tracking accuracy and efficiency.

Contribution of AGTP. Configuration #3 adds AGTP to the decoupled architecture (#2). Compared with Configuration #2, AO increases from 71.2% to 71.8%, while the inference speed decreases from 211 to 209 FPS. This gain indicates that AGTP effectively suppresses background-dominated template tokens and improves the discriminative quality of the target representation. Since AGTP is executed only once during template initialization and incurs negligible online overhead. These results demonstrate that template purification improves tracking accuracy with only a marginal impact on efficiency.

Table 3: Ablation study of PureTrack. We report AO on GOT-10k and inference speed on a single GPU. Checkmarks denote enabled components.

#	Decoupled	AGTP	$\tilde{M}_{1\to2}$	$\tilde{M}_{2\to1}$	BTC	s-t token	AO	FPS
1							70.6	162
2	✓						71.2	211
3	✓	✓					71.8	209
4	✓		✓				71.5	211
5	✓			✓			72.0	211
6	✓				✓		72.2	211
7	✓	✓			✓		72.3	209
8	✓	✓			✓	✓	**72.5**	208

Effectiveness of BTC Learning. Configurations #4 and #5 adopt single way temporal constraints ($\tilde{M}_{1\to2}$ and $\tilde{M}_{2\to1}$), yielding AO of 71.5% and 72.0%, respectively. The performance gap indicates complementary supervision between two directions, though neither single direction delivers optimal results. Combining both directions in configuration #6 raises AO to 72.2%, surpassing the decoupled baseline (#2) by 1.0% and the best one-way setting by 0.2%. And it brings no extra online computation and keeps inference speed at 211 FPS.

Complementarity between AGTP and BTC learning. Configuration # 7 integrates AGTP and BTC Loss. AO rises from 72.2% (# 6) to 72.3%, verifying their complementarity. AGTP enhances target representation discrimination, while the bidirectional loss regularizes temporal stability. The inference speed only drops slightly from 211 FPS to 209 FPS, indicating negligible overhead when combining both modules.

Necessity of the spatio-temporal token. We adopt the learnable spatio-temporal token in configuration #8. It achieves 72.5% AO, outperforming #7 by 0.2% and the baseline (#1) by 1.9%. This token efficiently aggregates and propagates global spatio-temporal information to complement temporal consistency supervision. It incurs negligible computation overhead, dropping speed merely by 1 FPS (209 FPS to 208 FPS). Finally, our full model boosts baseline AO from 70.6% to 72.5% and accelerates inference from 162 FPS to 208 FPS.

4.4 Qualitative Results

Visualization of AGTP. We visualize feature and response maps to explore the effectiveness of AGTP under typical challenging scenarios, including distractors, background clutter, occlusion, and fast motion. As illustrated in Figure 3(a), AGTP enables sufficient interactions among spatio-temporal and template tokens to adaptively refine template features, concentrating attention on the target and forming more discriminative representations. The response maps shown in Figure 3(b) further verifies this improvement. The model without AGTP generates misleading response peaks on interfering objects, while AGTP effectively suppresses background and distractor responses and produces accurate target response maps. These results confirm that AGTP strengthens target-related features, providing reliable priors for subsequent temporal consistency constraint optimization.

Visualization of BTC learning. We further verify the necessity of BTC via visualized response maps. In long-term tracking, appearance variation, similar distractors, occlusion, and fast motion easily cause cross-frame response drift. As shown in Figure 3(c), the model without this loss gradually drifts to distractors and suffers from localization failures under heavy occlusion, indicating poor temporal stability. By contrast, the BTC enforces dual temporal consistency and spatial alignment, yielding robust feature representations against motion variation and occlusion. It stably anchors response peaks to the true target and maintains precise localization in challenging scenarios.

Visualization of spatio-temporal token propagation. We visualize the attention distribution of the spatio-temporal token on representative LaSOT and TrackingNet sequences to analyze its long-term information propagation capability. As shown in Figure 4, the token consistently focuses on the target and adaptively shifts with target motion throughout tracking. It maintains compact target-oriented attention under background clutter, deformation, partial occlusion, and similar distractor interference, without obvious background diffusion. This demonstrates that the spatio-temporal token effectively

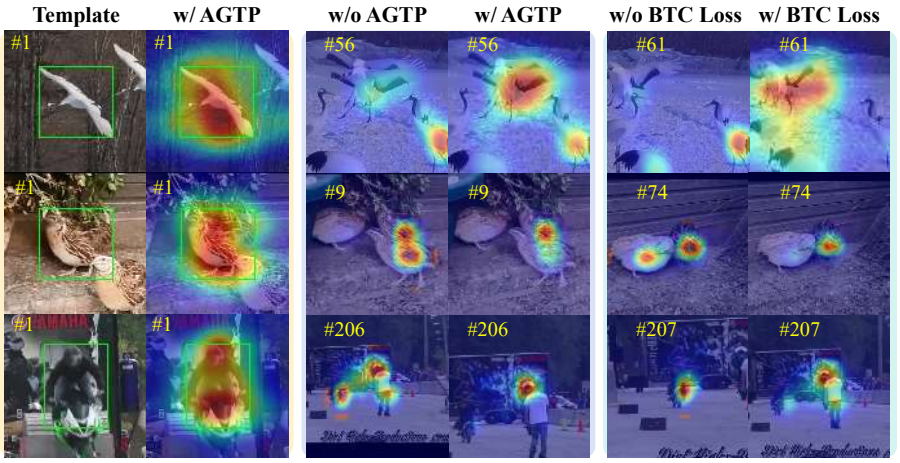


Figure 3: isualization on the GOT-10k dataset. (a) Attention maps between w/o and w AGTP. (b) Response maps between w/o and w AGTP. (c) Response maps between w/o and w BTC loss.

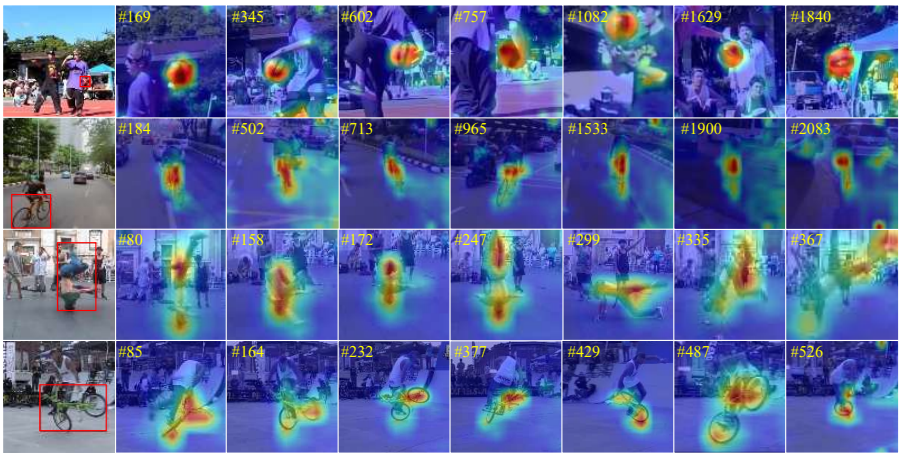


Figure 4: Visualization of attention maps of the spatio-temporal token during tracking on the LaSOT (first two sequences) and TrackingNet (third and fourth sequences) datasets.

preserves target identity and alleviates representation drift caused by appearance variations, while it provides reliable temporal information for dynamic feature updating and stable long-term tracking.

5 Conclusion

In this work, we propose PureTrack, an efficient framework that integrates template purification with temporal learning for reliable fixed-template visual tracking. Its decoupled architecture eliminates redundant template computation while enabling deep template encoding during initialization. Building upon this architecture, AGTP exploits spatio-temporal correlation cues to purify template representations, producing more focused and discriminative target features. Bidirectional temporal consistency learning further constrains the response maps of adjacent frames, improving localization stability under target motion, appearance variations, and occlusion. Meanwhile, spatio-temporal token serve as compact temporal state carriers that propagate target-relevant context across frames, compensating for the temporal information absent from fixed templates. PureTrack achieves a favorable balance between tracking accuracy and inference efficiency, providing a promising solution for reliable visual tracking in real-world applications.

AI USE STATEMENT

In this work, generative AI tools were used for paper writing polish and grammar correction. We have not used generative AI tools for designing core algorithm, conducting experiments or analyzing experimental results. All AI-assisted content has been carefully reviewed by the authors. The authors take full responsibility for all content of this paper.

REFERENCES

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

Luca Bertinetto, Jack Valmadre, Joao F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, 2016.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019.

Philippe Blatter, Menelaos Kanakis, Martin Danelljan, and Luc Van Gool. Efficient visual tracking with exemplar transformers. In *WACV*, 2023.

Vasyl Borsuk, Roman Vei, Orest Kupyn, Tetiana Martyniuk, Igor Krashenyi, and Jiri Matas. Fear: Fast, efficient, accurate and robust visual tracker. In *ECCV*, 2022.

Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. Robust object modeling for visual tracking. In *ICCV*, 2023.

Ziang Cao, Ziyuan Huang, Liang Pan, Shiwei Zhang, Ziwei Liu, and Changhong Fu. Tctrack: Temporal contexts for aerial tracking. In *CVPR*, 2022.

Ziang Cao, Ziyuan Huang, Liang Pan, Shiwei Zhang, Ziwei Liu, and Changhong Fu. Towards real-world visual tracking with temporal contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15834–15849, 2023.

Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021.

Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *CVPR*, 2023.

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*, 2025.

Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *CVPR*, 2022.

Yutao Cui, Tianhui Song, Gangshan Wu, and Limin Wang. Mixformerv2: Efficient fully transformer tracking. In *NeurIPS*, 2023.

Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, 2019.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *CVPR*, 2019.

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*, 2019.

Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, Yong Xu, Chunyuan Liao, Lin Yuan, and Haibin Ling. Lasot: A high-quality large-scale single object tracking benchmark. *IJCV*, 2020.

Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *CVPR*, 2021.

Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. Aiatrack: Attention in attention for transformer visual tracking. In *ECCV*, 2022.

Goutam Yelluru Gopal and Maria A. Amer. Separable self and mixed attention transformers for efficient object tracking. In *WACV*, 2024.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *PAMI*, 2019.

Ben Kang, Xin Chen, Dong Wang, Houwen Peng, and Huchuan Lu. Exploring lightweight hierarchical vision transformers for efficient visual tracking. In *ICCV*, 2023.

Ben Kang, Xin Chen, Simiao Lai, Yang Liu, Yi Liu, and Dong Wang. Mcitrack: Exploring enhanced contextual information for video-level object tracking. In *AAAI*, 2025.

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*, 2017.

Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.

Shuiwang Li, Yangxiang Yang, Dan Zeng, and Xucheng Wang. Adaptive and background-aware vision transformer for real-time uav tracking. In *ICCV*, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Hang Liu, Detian Huang, and Mingxin Lin. Fetrack: Feature-enhanced transformer network for visual object tracking. *Applied Sciences*, 14(22):10589, 2024.

Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *ICCV*, 2021.

Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016.

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*, 2018.

Minho Park, Jinjoo Song, and Sang Min Yoon. Visual object tracking using learnable target-aware token emphasis. *Engineering Applications of Artificial Intelligence*, 149:110482, 2025.

Liang Peng, Junyuan Gao, Xinran Liu, Weihong Li, Shaohua Dong, Zhipeng Zhang, Heng Fan, and Libo Zhang. Vasttrack: Vast category visual object tracking. *NeurIPS*, 2024.

Qiuhong Shen, Lei Qiao, Jinyang Guo, Peixia Li, Xin Li, Bo Li, Weitao Feng, Weihao Gan, Wei Wu, and Wanli Ouyang. Unsupervised learning of accurate siamese tracking. In *CVPR*, 2022.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

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

Qingmao Wei, Bi Zeng, Jianqi Liu, Li He, and Guotian Zeng. Litetrack: Layer pruning with asynchronous feature extraction for lightweight and efficient visual tracking. In *ICRA*, 2024.

Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *CVPR*, 2023.

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

You Wu, Yongxin Li, Mengyuan Liu, Xucheng Wang, Xiangyang Yang, Hengzhou Ye, Dan Zeng, Qijun Zhao, and Shuiwang Li. Learning adaptive and view-invariant vision transformer with multi-teacher knowledge distillation for real-time uav tracking. *arXiv*, 2024.

Liang Xu, Zhiqing Guo, and Liejun Wang. Efficient hybrid linear self-attention based visual object tracking with lora. *Neurocomputing*, 2025.

Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, 2020.

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*, 2025.

Chaocan Xue, Qihua Liang, Bineng Zhong, Yanting Zu, Yuanliang Xue, Haiying Xia, and Shuxiang Song. Toward low-cost yet effective temporal learning for uav tracking. In *CVPR*, 2026.

Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, 2021a.

Bin Yan, Houwen Peng, Kan Wu, Dong Wang, Jianlong Fu, and Huchuan Lu. Lighttrack: Finding lightweight neural networks for object tracking via one-shot architecture search. In *CVPR*, 2021b.

Dawei Yang, Jianfeng He, Yinchao Ma, Qianjin Yu, and Tianzhu Zhang. Foreground-background distribution modeling transformer for visual object tracking. In *ICCV*, pp. 10117–10127, 2023.

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*, 2022.

Hong Zhang, Wanli Xing, Huakao Lin, Hanyang Liu, and Yifan Yang. Efficient template distinction modeling tracker with temporal contexts for aerial tracking. *IEEE Trans. Geosci. Remote Sens.*, 62:1–11, 2024.

Hong Zhang, Jianbo Song, Hanyang Liu, Yang Han, Yifan Yang, and Huimin Ma. Awaretrack: Object awareness for visual tracking via templates interaction. *Image and Vision Computing*, 154: 105363, 2025.

Jinpu Zhang, Kaiheng Dai, Ziwen Li, Ruonan Wei, and Yuehuan Wang. Spatio-temporal matching for siamese visual tracking. *Neurocomputing*, 2023.

Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, 2019.

Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. Odtrack: Online dense temporal token learning for visual tracking. In *AAAI*, 2024.

Jiawen Zhu, Hao Tang, Xin Chen, Xiaolin Wang, Dong Wang, and Huchuan Lu. Two-stream beats one-stream: Asymmetric siamese network for efficient visual tracking. In *AAAI*, 2025.