RELITRACK: PSEUDO-STATE RELIABILITY LEARNING FOR SELF-SUPERVISED VISUAL TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised visual tracking reduces reliance on dense bounding-box annotations by deriving training signals from unlabeled videos. A key challenge, however, is that model-generated target states are not equally reliable: inaccurate pseudo-states can corrupt the target context used for subsequent self-supervised optimization. To address this issue, we propose ReliTrack, a reliability-aware self-supervised tracking method designed to mitigate *pseudo-state contamination*. ReliTrack first generates a forward pseudo-trajectory and estimates its reliability from the sharpness of the corresponding localization responses. Concentrated responses indicate confident target localization, whereas flat or ambiguous responses suggest unreliable pseudo-states. The resulting frame-level estimates are aggregated into a trajectory-level reliability score, which is converted into a conservative soft weight to regulate backward localization learning. This weighting suppresses contaminated supervision without discarding training samples. The reliability estimation requires neither additional annotations nor auxiliary networks and introduces no inference-time computation. Experiments demonstrate consistent improvements in tracking performance, while further analysis shows that ReliTrack shifts the effective training signal toward higher-quality pseudo-trajectories and reduces the contribution of inaccurate ones.

1 INTRODUCTION

Visual object tracking aims to continuously localize an arbitrary target throughout a video given its initial state. Recent progress has been driven by increasingly expressive architectures, ranging from Siamese networks to transformer-based trackers (Bertinetto et al., 2016; Li et al., 2019; Yan et al., 2021; Ye et al., 2022; Xing et al., 2023; Zheng et al., 2024). Despite their strong performance, these methods typically rely on large-scale datasets with dense frame-level bounding-box annotations, such as LaSOT, TrackingNet, GOT-10k, and COCO (Fan et al., 2019; Müller et al., 2018; Huang et al., 2021; Lin et al., 2014). Acquiring such annotations is costly, as the target must be identified and localized across the entire video. This reliance not only limits the scale and diversity of available training data but also hinders tracking models from exploiting abundant unlabeled videos.

Self-supervised visual tracking offers an alternative by deriving supervisory signals from the temporal structure of unlabeled videos. Existing approaches learn target correspondence through cycle consistency, synthetic target transformations, local region matching, or contrastive representation learning (Wang et al., 2019; Sio et al., 2020; Yuan et al., 2020a; Li et al., 2023; Yuan et al., 2020b). More recently, SSTrack (Zheng et al., 2025) introduced a forward–backward learning paradigm, where the tracker first localizes the target across unlabeled search frames and then reuses the predicted target states as contextual cues for backward grounding. While this paradigm enables tracking representations to be learned without dense annotations, it also makes subsequent self-supervised optimization highly dependent on the reliability of these model-generated intermediate states.

Forward tracking is inherently uncertain under occlusion, fast motion, deformation, and distractor interference, causing predicted target states to vary substantially in localization accuracy. Once an inaccurate state is used to crop a target region or construct target memory for backward grounding, the resulting localization objective is optimized under corrupted target context. Moreover, conventional sample averaging allows trajectories containing unreliable states to influence optimization as much as those with accurate localization. We refer to this failure mode as *pseudo-state contamina-*

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.

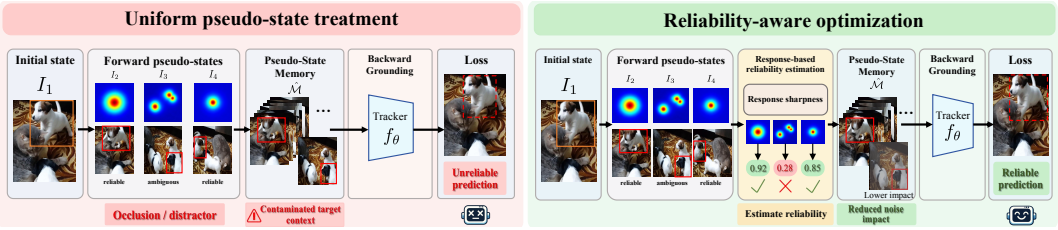


Figure 1: Motivation for reliability-aware optimization. Uniform treatment allows ambiguous or drifted pseudo-states to contaminate the target context, resulting in unreliable backward localization. In contrast, response-based reliability estimation reduces the influence of unreliable pseudo-states, leading to more reliable optimization and prediction.

tion. Unlike static annotation noise, pseudo-state contamination is generated by the tracker itself, evolves with the model during training, and propagates through temporal self-supervision.

Existing self-supervised objectives do not explicitly account for pseudo-state contamination. They generally lack a mechanism to assess the reliability of model-generated intermediate states and therefore treat reliable and inaccurate states similarly during optimization. Consequently, corrupted target context can directly affect the localization objective without being properly identified or suppressed. Addressing this issue requires a reliability signal that is available during training, reflects the quality of model-generated states, and does not depend on ground-truth boxes for intermediate frames.

To this end, we examine the localization responses produced during forward tracking. We observe that accurate pseudo-states typically yield responses concentrated around a dominant target location, whereas inaccurate states tend to produce flatter or more ambiguous responses with competing peaks. This observation motivates using response concentration as an internal estimate of pseudo-state reliability. Specifically, we characterize each response map by its standardized peak prominence, which we term *response sharpness*. By measuring how strongly the maximum response stands out from the overall response distribution, this statistic provides a label-free cue for distinguishing confident localization from ambiguous target assignment.

Based on this insight, we propose ReliTrack, a reliability-aware self-supervised tracking method. ReliTrack first generates a forward pseudo-trajectory and computes response sharpness for each intermediate prediction. These frame-level scores are converted into bounded confidence estimates and aggregated into a trajectory-level reliability score. This score then modulates the backward localization objective through conservative soft weighting, assigning larger weights to reliable trajectories while suppressing ambiguous ones without discarding them. The weights are normalized within each mini-batch to preserve the scale of the optimization objective and are activated only after an initial warm-up stage, when the response estimates become sufficiently stable. Since the required response maps are already produced during forward localization, ReliTrack requires neither an auxiliary reliability network nor additional inference-time computation.

Experiments on GOT-10k (Huang et al., 2021), LaSOT (Fan et al., 2019), TrackingNet (Müller et al., 2018), and LaSOT$_{\text{ext}}$ (Fan et al., 2021) demonstrate consistent improvements over the reproduced self-supervised baseline. Beyond final tracking accuracy, we further analyze how reliability weighting affects the supervision received by the optimization objective, showing that it increases the effective quality of training trajectories while suppressing the influence of inaccurate ones throughout training. These results demonstrate the effectiveness of ReliTrack in improving self-supervised tracking and mitigating pseudo-state contamination. Our contributions are summarized as follows:

1. We propose a response-sharpness-based reliability estimation strategy that quantifies the quality of model-generated pseudo-states from their localization responses without requiring intermediate-frame annotations.
2. We propose ReliTrack, a reliability-aware self-supervised tracker that aggregates frame-level confidence across the generated trajectory and regulates backward localization learning through conservative soft weighting.
3. Experimental results on four tracking benchmarks, together with pseudo-state quality and effective-noise analyses, demonstrate consistent performance improvements and validate the effectiveness of ReliTrack.

2 Related Work

Visual object tracking has been extensively studied under the fully supervised setting. Early Siamese trackers formulate tracking as template-search matching and learn generic target similarity from annotated image pairs (Bertinetto et al., 2016; Li et al., 2019). Discriminative tracking methods further improve localization by learning target-specific prediction models or overlap-aware objectives (Danelljan et al., 2019; Bhat et al., 2019). More recently, transformer-based trackers jointly model target and search features through attention mechanisms, enabling stronger feature interaction and more expressive target localization (Yan et al., 2021; Ye et al., 2022; Cui et al., 2022; Xing et al., 2023; Chen et al., 2023; Zheng et al., 2024). Beyond pairwise template-search matching, recent trackers further exploit temporal memory, historical target states, or sequential prediction to improve robustness against appearance variation and distractors (Fu et al., 2021; Mayer et al., 2021; Cai et al., 2024; Bai et al., 2024). These advances have substantially improved tracking accuracy across short- and long-term benchmarks. However, their training typically relies on large-scale datasets with dense bounding-box annotations (Fan et al., 2019; Müller et al., 2018; Huang et al., 2021; Lin et al., 2014), whose annotation cost limits the diversity of target categories and video domains available for learning. In contrast to these architectural advances, our work focuses on the reliability of pseudo supervision when dense annotations are unavailable during training.

Self-supervised visual tracking seeks to learn tracking representations from unlabeled or weakly annotated videos. One line of work exploits temporal correspondence and cycle consistency. Temporal cycle consistency has been widely studied for video representation learning (Wang et al., 2019) and later adapted to tracking through cycle-consistent Siamese frameworks (Yuan et al., 2020b). Other approaches construct self-supervised tracking samples through synthetic target transformations or local region matching. S^2SiamFC learns Siamese tracking without dense labels by sampling image regions and their enlarged contexts (Sio et al., 2020); Self-SDCT introduces self-supervised correlation learning for deep correlation tracking (Yuan et al., 2020a); and TADS synthesizes target-aware training data for self-supervised tracking (Li et al., 2023). Contrastive objectives provide an additional supervisory signal by encouraging correspondence between different views of the same instance (Oord et al., 2018; He et al., 2020; Chen & He, 2021). More recently, SSTrack combines forward target localization with backward grounding and instance-level representation learning (Zheng et al., 2025). Its forward predictions are reused to construct target context for subsequent grounding, enabling learning from videos without dense frame-level annotations. However, these intermediate target states remain model-generated predictions and may vary substantially in localization quality. Existing objectives, including cycle consistency and contrastive learning, do not explicitly account for how pseudo trajectories with different reliability should contribute to localization learning. Our work addresses this limitation by estimating reliability from the response maps of forward pseudo-states and using it to reweight the resulting localization objective.

3 Methodology

3.1 Preliminaries

Given an unlabeled video sequence $\mathcal{V} = \{I_t\}_{t=1}^{T}$ and an initial target box b_1, self-supervised visual tracking aims to train a tracker without dense annotations for subsequent frames. We denote the target state at frame t as $s_t = (I_t, b_t)$, where I_t denotes the frame and b_t the corresponding target bounding box. During training, only the initial state $s_1 = (I_1, b_1)$ is available.

Forward pseudo-state generation. Let f_θ denote the tracking network. Starting from the initial state s_1, the tracker is sequentially applied to each subsequent search frame I_t, where $t = 2, \ldots, T_s$, to predict a pseudo target state:

$$(\hat{b}_t, R_t) = f_\theta(I_t, \mathcal{M}_{1:t-1}), \quad \hat{s}_t = (I_t, \hat{b}_t), \tag{1}$$

where $\mathcal{M}_{1:t-1}$ denotes the target memory constructed from the initial state and previously generated pseudo-states, $\hat{b}_t \in \mathbb{R}^4$ is the predicted bounding box, and $R_t \in \mathbb{R}^{H \times W}$ is the corresponding response map. Repeating this process over the forward search frames yields the pseudo-trajectory

$$s_1 \to \hat{s}_2 \to \cdots \to \hat{s}_{T_s}. \tag{2}$$

Since these generated states are not supervised by dense annotations, their localization quality may vary across frames. Let $b_t^\star$ denote the latent ground-truth box that would be available under full

supervision. The predicted pseudo box can then be expressed as

$$\hat{b}_t = b_t^\star + \epsilon_t, \tag{3}$$

where ϵ_t denotes the localization error. Although ϵ_t is unobservable during training, it directly affects the reliability of the pseudo-state and any subsequent supervision constructed from it.

Backward self-supervision. The generated pseudo-states $\{\hat{s}_t\}_{t=2}^{T_s}$ are then used as target context for backward grounding. Given the initial frame or its augmented view I_1^a with known target box b_1^a, the tracker predicts a recovered target state:

$$(\tilde{b}_1, \tilde{R}_1) = f_\theta(I_1^a, \hat{\mathcal{M}}), \tag{4}$$

where $\hat{\mathcal{M}}$ denotes the target memory constructed from $\{\hat{s}_t\}_{t=2}^{T_s}$. This forms a closed self-supervised path

$$s_1 \rightarrow \{\hat{s}_t\}_{t=2}^{T_s} \rightarrow \tilde{s}_1, \tag{5}$$

where $\tilde{s}_1 = (I_1^a, \tilde{b}_1)$ is expected to recover the known target state. The supervisory signal is therefore obtained by comparing the recovered box $\tilde{b}_1$ with b_1^a, without requiring dense annotations for the intermediate frames.

Tracking loss. For each recovered target state, the localization objective consists of bounding-box regression and response-map localization terms:

$$\ell_{\text{trk}}^i = \alpha_{\text{giou}}\ell_{\text{giou}}^i + \alpha_1\ell_1^i + \alpha_{\text{focal}}\ell_{\text{focal}}^i. \tag{6}$$

where ℓ_{giou}^i and ℓ_1^i are computed between $\tilde{b}_1$ and b_1^a, while ℓ_{focal}^i supervises the response map. Existing self-supervised objectives typically aggregate samples without accounting for the reliability of their pseudo-trajectories:

$$\mathcal{L}_{\text{ssl}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\ell_{\text{trk}}^i. \tag{7}$$

Although the loss is evaluated against the known target box b_1^a, the target context used for backward grounding is constructed from forward pseudo-states. Consequently, localization errors in these pseudo-states can propagate into the backward supervision, while uniform averaging fails to distinguish trajectories of different reliability.

3.2 Analysis

Pseudo-states with large localization errors may introduce incorrect target context into the self-supervised path, a failure mode we term *pseudo-state contamination*. Unlike fixed external label noise, such contamination is generated by the tracker itself. Once an inaccurate pseudo-state is incorporated into the target memory for backward grounding, the resulting loss is optimized under corrupted target context. To mitigate this problem, self-supervised samples with pseudo-trajectories of different quality should not contribute equally to the optimization objective. Let $q_i \in [0, 1]$ denote the estimated reliability of the forward pseudo-trajectory associated with sample i, where a larger value indicates higher reliability. We use this estimate to assign a weight $w(q_i)$ to the corresponding backward localization loss:

$$\mathcal{L}_w(\theta) = \frac{\sum_{i=1}^{N} w(q_i)\ell_{\text{trk}}^i}{\sum_{i=1}^{N} w(q_i)}, \tag{8}$$

where $w(q_i)$ denotes a non-negative reliability weight. The normalization maintains a stable loss scale despite variations in the weight distribution across mini-batches.

Inspired by the Peak-to-Sidelobe Ratio (PSR) used to characterize response confidence in correlation-filter tracking (Bolme et al., 2010), we examine the response maps produced during forward tracking to estimate q_i before backward supervision. We observe a consistent relationship between response concentration and localization quality. A response concentrated around a dominant peak typically corresponds to an accurate pseudo box, whereas a flat response or multiple competing peaks often indicate ambiguous localization and a higher risk of drift. Figure 2 illustrates this relationship. This observation motivates relating reliability to the underlying pseudo-state error.

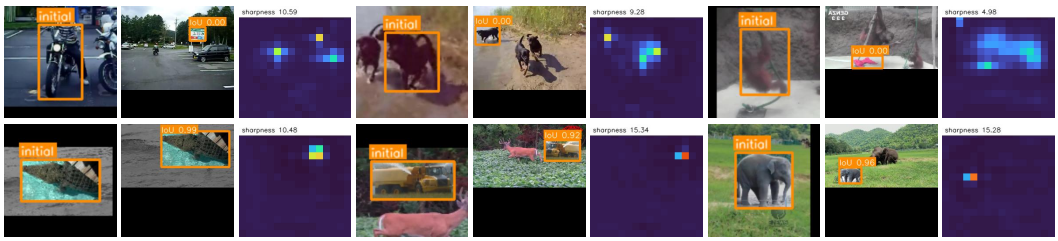


Figure 2: Illustration of the relationship between response-map concentration and pseudo-state quality. Concentrated responses usually correspond to reliable pseudo states, while flat or multi-modal responses are more likely to produce inaccurate intermediate predictions.

Let $d(\cdot,\cdot)$ denote a localization error measure. For a trajectory, we define its average localization error as $D = (T_s-1)^{-1}\sum_{t=2}^{T_s} d(\hat{b}_t, b_t^\star)$. The effective trajectory error after weighting is

$$\mathcal{E}_w = \frac{\mathbb{E}\left[w(q)D\right]}{\mathbb{E}\left[w(q)\right]}. \tag{9}$$

Using the covariance identity, $\mathcal{E}_w$ can be decomposed as

$$\mathcal{E}_w = \mathbb{E}\left[D\right] + \frac{\mathrm{Cov}\left(w(q), D\right)}{\mathbb{E}\left[w(q)\right]}. \tag{10}$$

Relative to the unweighted error $\mathbb{E}[D]$, reliability weighting reduces the effective trajectory error when $\mathrm{Cov}(w(q), D) < 0$. This provides a concrete criterion for designing the reliability measure.

3.3 ReliTrack

Figure 3 presents an overview of ReliTrack. Given an initial target state and unlabeled search frames, ReliTrack performs reliability-aware self-supervised learning by generating forward pseudo-states, estimating their trajectory-level reliability from localization responses, and using it to regulate backward localization learning. Specifically, the latent quality of a forward pseudo-trajectory is mapped to a training weight as

$$R_t \longmapsto \kappa_t \longmapsto q \longmapsto w(q), \tag{11}$$

where R_t denotes the response map of the t-th forward prediction, κ_t the corresponding frame-level response sharpness, q the trajectory-level reliability aggregated from bounded confidence scores, and $w(q)$ the weight applied to the backward localization objective.

Response sharpness reliability. Let Ω denote the spatial domain of the response map, containing all candidate locations on the response grid. For $t = 2, \ldots, T_s$ and $u \in \Omega$, $R_t(u)$ denotes the detached, sigmoid-activated center response at location u in frame I_t, without additional spatial normalization. We characterize each response map using its maximum p_t, spatial mean μ_t, and spatial standard deviation σ_t:

$$p_t = \max_{u\in\Omega} R_t(u), \quad \mu_t = \frac{1}{|\Omega|}\sum_{u\in\Omega} R_t(u), \tag{12}$$

$$\sigma_t = \sqrt{\frac{1}{|\Omega|}\sum_{u\in\Omega}\left(R_t(u)-\mu_t\right)^2}. \tag{13}$$

Here, p_t captures the strongest target evidence, while μ_t and σ_t characterize the overall response level and dispersion, respectively. Reliable localization is expected to exhibit a dominant peak that clearly stands out from the overall response distribution. We therefore define response sharpness as

$$\kappa_t = \frac{p_t - \mu_t}{\sigma_t + \varepsilon}, \tag{14}$$

where ε is a small constant for numerical stability. The numerator measures the prominence of the maximum response relative to the mean, while the denominator normalizes this prominence by the

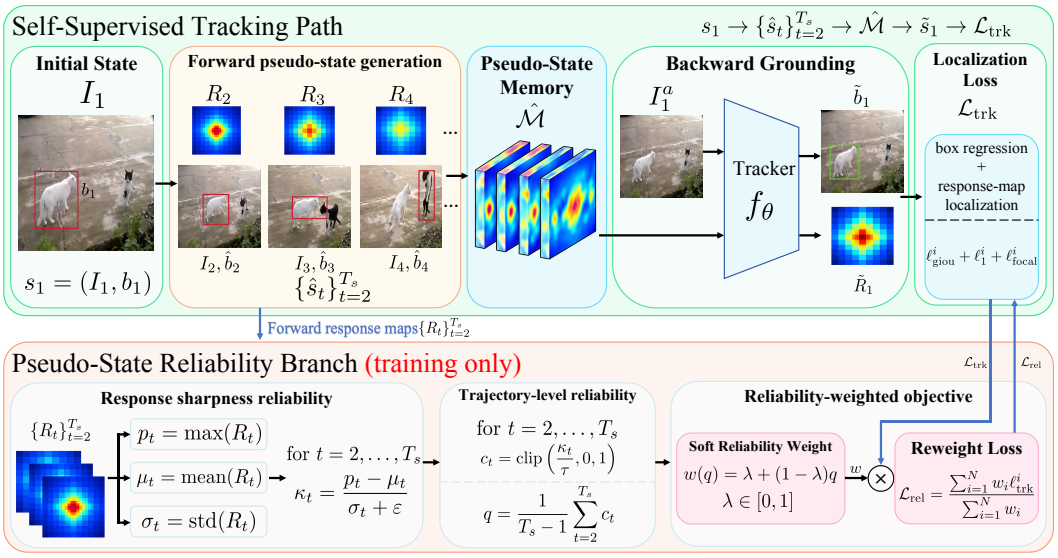


Figure 3: Overview of ReliTrack. Forward tracking generates a pseudo trajectory whose states provide target context for backward grounding. Response sharpness is measured for each generated state and aggregated into trajectory-level reliability, which regulates the backward localization objective. Reliability-aware learning is used only during training.

spatial response variation. Consequently, a larger κ_t indicates a more dominant response peak and thus more confident localization, whereas a smaller κ_t reflects a flatter or more ambiguous response distribution.

Trajectory-level reliability. A training sample may contain multiple forward search frames and thus multiple generated pseudo-states. Let T_s denote the final frame index of the forward trajectory. Excluding the observed initial state s_1, the trajectory contains $T_s - 1$ pseudo-states indexed by $t = 2, \ldots, T_s$. Since the raw sharpness values are unbounded, each frame-level score is first mapped to a bounded confidence:

$$c_t = \text{clip}\left(\frac{\kappa_t}{\tau}, 0, 1\right), \tag{15}$$

where τ is a scale parameter and $\text{clip}(\cdot, 0, 1)$ restricts the output to $[0, 1]$. The trajectory-level reliability q is then obtained by aggregating the confidence scores over all generated pseudo-states:

$$q = \text{Agg}_{t=2}^{T_s} c_t. \tag{16}$$

In our implementation, Agg is the mean operator, yielding

$$q = \frac{1}{T_s - 1} \sum_{t=2}^{T_s} c_t. \tag{17}$$

Thus, q reflects the average response confidence along the forward pseudo-trajectory. While other aggregation rules are compatible with this formulation, mean aggregation provides a stable summary of multiple generated states.

Reliability-weighted objective. The trajectory-level reliability q is converted into a conservative soft weight:

$$w(q) = \lambda + (1 - \lambda)q, \quad \lambda \in [0, 1]. \tag{18}$$

where λ denotes the minimum weight. Since $q \in [0, 1]$, the resulting weight satisfies $w(q) \in [\lambda, 1]$. This soft weighting avoids discarding low-confidence pseudo-states; instead, unreliable samples are down-weighted while remaining available for self-supervised learning.

The final reliability-weighted localization objective is

$$\mathcal{L}_{\text{rel}} = \frac{\sum_{i=1}^N w_i \left(\alpha_{\text{giou}} \ell^i_{\text{giou}} + \alpha_1 \ell^i_1 + \alpha_{\text{focal}} \ell^i_{\text{focal}}\right)}{\sum_{i=1}^N w_i}, \quad w_i = w(q_i). \tag{19}$$

Here, N denotes the mini-batch size, q_i the trajectory-level reliability of the i-th sample, and w_i its corresponding loss weight. The terms ℓ^i_{giou}, ℓ^i_1, and ℓ^i_{focal} are the per-sample losses defined in Eq. 6, with coefficients α_{giou}, α_1, and α_{focal}, respectively. Normalizing by the total sample weight maintains a stable objective scale across mini-batches with different reliability distributions.

Reliability-aware weighting is applied only during training. Since the required response maps are already produced during forward localization, ReliTrack requires no auxiliary reliability network and incurs no additional inference cost. To avoid unreliable estimates from unstable early responses, reliability weighting is activated only after an initial warm-up stage.

4 Experiments

4.1 Experimental Setup

For evaluation on the GOT-10k test split, ReliTrack is trained exclusively on the GOT-10k training split, without using LaSOT, TrackingNet, or COCO; for the other benchmarks, training samples are drawn from LaSOT, GOT-10k, COCO, and TrackingNet (Fan et al., 2019; Huang et al., 2021; Lin et al., 2014; Müller et al., 2018). ReliTrack is initialized with DropMAE and only the initial target annotation of each training sample is used while training. The search and template resolutions are 256×256 and 128×128, respectively. Each training sample consists of one template frame, three forward search frames, and two grounding frames. We train the model for 150 epochs using AdamW with a per-GPU batch size of 16. The initial learning rate is 2.5×10^{-4} with a backbone multiplier of 0.1 and a weight decay of 10^{-4}. The learning rate is reduced by a factor of 0.1 after epoch 120, and the gradient norm is clipped at 0.1. The weights for the GIoU, ℓ_1, and focal losses are set to 2, 5, and 1, respectively. Reliability weighting is activated at epoch 21 following a 20-epoch warm-up. Unless otherwise specified, we use mean trajectory aggregation with $\lambda = 0.8$ and $\tau = 16$.

Table 1: Comparison with state-of-the-art trackers on GOT10K, LaSOT, TrackingNet, and $\text{LaSOT}_{\text{ext}}$. Best and second-best self-supervised results are shown in red and blue.

Type	Method	GOT10K*			LaSOT			TrackingNet			$\text{LaSOT}_{\text{ext}}$		
		AO	$\text{SR}_{0.5}$	$\text{SR}_{0.75}$	AUC	P_{Norm}	P	AUC	P_{Norm}	P	AUC	P_{Norm}	P
Fully Sup	SiamRPN++ (Li et al., 2019)	51.7	61.6	32.5	49.6	56.9	49.1	73.3	80.0	69.4	34.0	41.6	39.6
	DiMP (Bhat et al., 2019)	61.1	71.7	49.2	56.9	65.0	56.7	74.0	80.1	68.7	39.2	47.6	45.1
	SiamRCNN (Voigtlaender et al., 2020)	64.9	72.8	59.7	64.8	72.2	-	81.2	85.4	80.0	-	-	-
	Ocean (Zhang et al., 2020)	61.1	72.1	47.3	56.0	65.1	56.6	-	-	-	-	-	-
	STMTrack (Fu et al., 2021)	64.2	73.7	57.5	60.6	69.3	63.3	80.3	85.1	76.7	-	-	-
	TransT (Chen et al., 2021)	67.1	76.8	60.9	64.9	73.8	69.0	81.4	86.7	80.3	-	-	-
	STARK (Yan et al., 2021)	68.8	78.1	64.1	67.1	77.0	-	82.0	86.9	-	-	-	-
	KeepTrack (Mayer et al., 2021)	-	-	-	67.1	77.2	70.2	-	-	-	48.2	-	-
	MixFormer (Cui et al., 2022)	70.7	80.0	67.8	69.2	78.7	74.7	83.1	88.1	81.6	-	-	-
	TransInMo (Guo et al., 2022)	-	-	-	65.7	76.0	70.7	81.7	-	-	-	-	-
	OSTrack (Ye et al., 2022)	73.7	83.2	70.8	71.1	81.1	77.6	83.9	88.5	83.2	50.5	61.3	57.6
	AiATrack (Gao et al., 2022)	69.6	80.0	63.2	69.0	79.4	73.8	82.7	87.8	80.4	47.7	55.6	55.4
	SeqTrack (Chen et al., 2023)	74.5	84.3	71.4	71.5	81.1	77.8	83.9	88.8	83.6	50.5	61.6	57.5
	GRM (Gao et al., 2023)	73.4	82.9	70.4	69.9	79.3	75.8	84.0	88.7	83.3	-	-	-
	VideoTrack (Xie et al., 2023)	72.9	81.9	69.8	70.2	-	76.4	83.8	88.7	83.1	-	-	-
	ARTrack (Xing et al., 2023)	75.5	84.3	74.3	72.6	81.7	79.1	85.1	89.1	84.8	51.9	62.0	58.5
	ODTrack (Zheng et al., 2024)	77.0	87.9	75.1	73.2	83.2	80.6	85.1	90.1	84.9	52.4	63.9	60.1
	HIPTrack (Cai et al., 2024)	77.4	88.0	74.5	72.7	82.9	79.5	84.5	89.1	83.8	-	-	-
	AQATrack (Xie et al., 2024)	76.0	85.2	74.9	72.7	82.9	80.2	84.8	89.3	84.3	52.7	64.2	60.8
	ARTrackV2 (Bai et al., 2024)	77.5	86.0	75.5	73.0	82.0	79.6	85.7	89.8	85.5	52.9	63.4	59.1
	AsymTrack (Jiawen et al., 2025)	67.7	76.7	61.4	64.7	73.0	67.8	80.0	84.5	77.4	-	-	-
	SUTrack (Chen et al., 2025)	77.9	87.5	78.5	73.2	83.4	80.5	85.7	90.3	85.1	53.1	64.2	60.5
	DreamTrack (Guo et al., 2025)	77.5	87.1	74.2	73.8	83.4	80.6	85.8	90.0	85.3	53.1	64.1	59.8
	ARPTrack (Liang et al., 2025)	77.7	87.3	74.3	72.6	81.4	78.5	85.5	90.0	85.3	52.0	62.9	58.7
	ARTrack-AC (Lin et al., 2026)	72.3	82.4	67.3	66.7	76.5	71.5	81.8	86.5	78.4	47.5	57.9	53.3
Self Sup	TADS (Li et al., 2023)	46.7	56.5	21.1	45.5	54.2	44.8	65.6	73.4	60.6	-	-	-
	SSTrack (Zheng et al., 2025)	63.10	72.40	55.00	64.84	74.98	68.85	79.52	85.80	77.24	44.95	56.22	50.04
	ReliTrack (ours)	66.20	75.50	58.40	66.37	76.92	70.84	80.38	86.38	78.25	46.51	57.98	52.21

4.2 Main Results

We compare ReliTrack with representative fully supervised and self-supervised trackers on standard tracking benchmarks. GOT-10k follows a one-shot evaluation protocol with no overlap between the object classes in its training and test splits. LaSOT is a large-scale long-term tracking benchmark with long sequences and diverse target categories. It contains frequent appearance changes, scale variation, partial occlusion, and out-of-view cases, making it suitable for evaluating whether a tracker can maintain target identity over extended temporal spans. $\text{LaSOT}_{\text{ext}}$ further

extends the evaluation to additional object categories that are not covered by the original LaSOT benchmark. It places stronger emphasis on category-level generalization and distractor robustness, which is particularly relevant for self-supervised trackers trained without dense manual annotations. TrackingNet provides a large-scale short-term tracking evaluation collected from unconstrained web videos, where the target motion and scene composition are more diverse.

As shown in Tab. 1, ReliTrack outperforms SSTrack on all evaluated benchmarks. On GOT-10k, ReliTrack achieves 66.20 on AO, 75.50 on $SR_{0.5}$, and 58.40 on $SR_{0.75}$, surpassing SSTrack by 3.10, 3.10, and 3.40 points, respectively. On LaSOT, ReliTrack achieves 66.37 on AUC, 76.92 on P_{Norm}, and 70.84 on P, surpassing SSTrack by 1.53, 1.94, and 1.99 points, respectively. On TrackingNet, ReliTrack achieves 80.38 on AUC, 86.38 on P_{Norm}, and 78.25 on P, surpassing SSTrack by 0.86, 0.58, and 1.01 points, respectively. On $LaSOT_{ext}$, ReliTrack reaches 46.51 on AUC, 57.98 on P_{Norm}, and 52.21 on P, surpassing SSTrack by 1.56, 1.76, and 2.17 points, respectively. Taken together, these results demonstrate that reliability-aware learning consistently improves tracking performance across diverse evaluation settings, supporting the effectiveness and generalization of ReliTrack. Meanwhile, since reliability estimation and weighting are used only during self-supervised training, ReliTrack introduces no additional test-time computation.

4.3 Ablation Studies

We ablate three components that map response sharpness to pseudo-state weights: trajectory aggregation, the minimum weight λ, and the sharpness scale τ. The reliability cue remains fixed in all experiments. Table 2 summarizes the results. Mean aggregation performs best on LaSOT and remains competitive on $LaSOT_{ext}$; max aggregation is consistently lower, while min aggregation favors $LaSOT_{ext}$ at the cost of a 1.35-point AUC drop on LaSOT. For the soft weighting parameters, $\lambda = 0.80$ provides the strongest overall results, whereas stronger suppression at $\lambda = 0.75$ degrades both benchmarks. Similarly, $\tau = 16$ achieves the best balance across datasets: $\tau = 8$ performs better on $LaSOT_{ext}$ but worse on LaSOT, and larger values consistently reduce performance. We therefore use mean aggregation, $\lambda = 0.80$, and $\tau = 16$ in the final model.

Table 2: Ablation studies on trajectory aggregation, the minimum reliability weight λ, and the sharpness scale τ. Parentheses show differences from the adopted setting in each block.

Setting	LaSOT			$LaSOT_{ext}$		
	AUC	P_{Norm}	P	AUC	P_{Norm}	P
Trajectory aggregation						
Mean (adopted)	**66.37**	**76.92**	**70.84**	**46.51**	**57.98**	**52.21**
Max	66.14 (-0.23)	76.57 (-0.35)	70.65 (-0.19)	46.32 (-0.19)	57.68 (-0.30)	51.97 (-0.24)
Min	65.02 (-1.35)	75.28 (-1.64)	69.27 (-1.57)	46.73 (+0.22)	58.11 (+0.13)	52.87 (+0.66)
Minimum reliability weight λ						
0.80 (adopted)	**66.37**	**76.92**	**70.84**	**46.51**	**57.98**	**52.21**
0.90	65.34 (-1.03)	75.74 (-1.18)	69.70 (-1.14)	46.22 (-0.29)	57.74 (-0.24)	51.83 (-0.38)
0.85	65.65 (-0.72)	76.23 (-0.69)	70.32 (-0.52)	46.39 (-0.12)	57.86 (-0.12)	52.35 (+0.14)
0.75	64.97 (-1.40)	75.06 (-1.86)	69.03 (-1.81)	44.45 (-2.06)	55.54 (-2.44)	49.73 (-2.48)
Sharpness scale τ						
16 (adopted)	**66.37**	**76.92**	**70.84**	**46.51**	**57.98**	**52.21**
8	65.29 (-1.08)	75.76 (-1.16)	70.03 (-0.81)	46.86 (+0.35)	58.36 (+0.38)	52.84 (+0.63)
12	63.91 (-2.46)	74.30 (-2.62)	68.43 (-2.41)	45.29 (-1.22)	56.36 (-1.62)	51.28 (-0.93)
20	64.11 (-2.26)	74.28 (-2.64)	68.63 (-2.21)	45.01 (-1.50)	56.15 (-1.83)	50.71 (-1.50)
24	64.14 (-2.23)	74.53 (-2.39)	69.09 (-1.75)	44.40 (-2.11)	55.60 (-2.38)	50.12 (-2.09)

4.4 Analysis

We further examine whether response sharpness provides a valid reliability signal and whether the resulting weights improve the quality of the supervision received by the training objective. This analysis uses the ground-truth boxes associated with the sampled training frames only to measure

pseudo-state quality after the forward pass. These annotations are not accessed by the tracker or used to construct the reliability weights.

Response sharpness and pseudo-state quality. We first analyze the forward predictions produced by the unweighted SSTrack model, thereby avoiding the influence of reliability weighting on the response statistics. We measure pseudo-state quality by $o_t = \mathrm{IoU}(\hat{b}_t, b_t^\star)$ and divide all predictions into ten equally populated groups according to their response sharpness. As shown in Figure 4, mean pseudo-state IoU increases monotonically with response sharpness, while the proportion of inaccurate pseudo states with $o_t < 0.3$ decreases monotonically. In particular, mean pseudo-state IoU increases from 0.100 in the lowest-sharpness group to 0.898 in the highest-sharpness group, whereas the inaccurate-state ratio decreases from 86.35% to 1.30%. Across all predictions, response sharpness has a Pearson correlation of 0.681 and a Spearman rank correlation of 0.679 with pseudo-state IoU. When predictions with $o_t \geq 0.5$ are treated as reliable pseudo states, sharpness obtains an AUC of 0.868. These results establish that response sharpness preserves a strong ordering of pseudo-state quality although corresponding ground-truth state is unavailable to the weighting function.

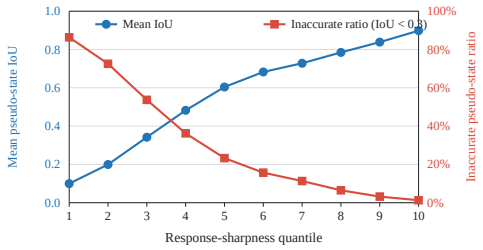


Figure 4: Response sharpness versus forward pseudo-state quality for the unweighted SSTrack model. Predictions are grouped into ten equal-frequency bins; ground-truth boxes are used only for post-hoc analysis.

Table 3: Effective pseudo-state quality before and after reliability weighting. Reliable and inaccurate trajectories are defined by IoU thresholds of ≥ 0.5 and < 0.3, respectively. Statistics are averaged within each mini-batch following Eq. 19; changes are percentage points.

Metric	Unweighted	Weighted	Change
Mean IoU	0.5993	**0.6098**	+1.05
Min IoU	0.5057	**0.5164**	+1.06
Last IoU	0.5916	**0.6021**	+1.05
Reliable ratio	67.22%	**68.47%**	+1.25
Inaccurate ratio	26.29%	**25.16%**	-1.13

Quality of the effective training signal. We next quantify how the proposed weighting changes the pseudo-state quality represented in the optimization objective. For trajectory i, let $\bar{o}_i = (T_s - 1)^{-1} \sum_{t=2}^{T_s} \mathrm{IoU}(\hat{b}_{i,t}, b_{i,t}^\star)$ be its mean pseudo-state IoU. Within each mini-batch, we compare the ordinary average $N^{-1} \sum_i \bar{o}_i$ with the effective quality after weighting,

$$\bar{o}_w = \frac{\sum_{i=1}^{N} w_i \bar{o}_i}{\sum_{i=1}^{N} w_i}. \tag{20}$$

Table 3 reports the results over 5,330 recorded mini-batches after reliability weighting is activated. The effective mean trajectory IoU increases from 0.5993 to 0.6098. The same behavior is observed when trajectory quality is measured by the least accurate state or the last generated state, for which weighting improves effective IoU by 1.06 and 1.05 points, respectively. Moreover, the effective proportion of reliable trajectories increases by 1.25 points, while the contribution of inaccurate trajectories decreases by 1.13 points. The mean trajectory IoU improvement remains consistent across training stages, reaching 1.06 points in epochs 21–40, 1.04 points in epochs 41–100, and 1.05 points in epochs 101–150. Therefore, the proposed weights consistently shift optimization toward more accurate pseudo trajectories, providing empirical support for the effective-noise analysis in Eq. 10.

5 Conclusion

We present ReliTrack, a reliability-aware self-supervised tracking method for mitigating pseudo-state contamination. By estimating frame-level response sharpness, aggregating it into trajectory-level reliability, and incorporating the resulting reliability into the localization objective, ReliTrack adaptively suppresses unreliable pseudo supervision while preserving useful training signals from unlabeled videos. Experiments on GOT-10k, LaSOT, TrackingNet, and $\mathrm{LaSOT}_{\mathrm{ext}}$ demonstrate consistent performance improvements. Further analyses show that ReliTrack improves the effective quality of pseudo trajectories, mitigates contamination during self-supervised optimization, and enhances overall tracking performance without additional inference-time computation.

AI USE STATEMENT

The authors used generative AI tools to assist with language polishing and grammar checking. All AI-assisted content was reviewed and revised by the authors. The authors take full responsibility for the final content of the paper, including all text, claims, experiments, and code.

ETHICS STATEMENT

This work uses publicly available datasets for visual tracking research and does not involve human subject experiments, private personal data, or sensitive attributes. We do not foresee direct negative societal impacts beyond the general dual-use considerations associated with visual tracking technologies.

REPRODUCIBILITY STATEMENT

We provide implementation details, training settings, dataset configurations, evaluation protocols, and ablation studies in the main paper. Code and configuration files will be released upon publication to facilitate reproducibility.

REFERENCES

Yifan Bai, Zeyang Zhao, Yihong Gong, and Xing Wei. ARTrackV2: Prompting autoregressive tracker where to look and how to describe. In *CVPR*, pp. 19048–19057, 2024.

Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, pp. 850–865, 2016.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pp. 6181–6190, 2019.

David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, 2010. doi: 10.1109/CVPR.2010.5539960.

Wenrui Cai, Qingjie Liu, and Yunhong Wang. Hiptrack: Visual tracking with historical prompts. In *CVPR*, pp. 19258–19267, 2024.

Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, pp. 8126–8135, 2021.

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

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, pp. 15750–15758, 2021.

Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *CVPR*, pp. 13598–13608, 2022.

Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate tracking by overlap maximization. In *CVPR*, pp. 4660–4669, 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*, pp. 5374–5383, 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. *International Journal of Computer Vision*, pp. 439–461, 2021.

Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *CVPR*, pp. 13774–13783, 2021.

Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. Aiatrack: Attention in attention for transformer visual tracking. In *ECCV*, pp. 146–164, 2022.

Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. *CVPR*, 2023.

Mingzhe Guo, Zhipeng Zhang, Heng Fan, Liping Jing, Yilin Lyu, Bing Li, and Weiming Hu. Learning target-aware representation for visual tracking via informative interactions. In *IJCAI*, pp. 927–934, 2022.

Mingzhe Guo, Weiping Tan, Wenyu Ran, Liping Jing, and Zhipeng Zhang. Dreamtrack: Dreaming the future for multimodal visual object tracking. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7201–7210. IEEE, 2025.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pp. 9729–9738, 2020.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1562–1577, 2021.

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

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*, pp. 4282–4291, 2019.

Xin Li, Wenjie Pei, Yaowei Wang, Zhenyu He, Huchuan Lu, and Ming-Hsuan Yang. Self-supervised tracking via target-aware data synthesis. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Shiyi Liang, Yifan Bai, Yihong Gong, and Xing Wei. Autoregressive sequential pretraining for visual tracking. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7254–7264. IEEE, 2025.

Tong Lin, Yifan Bai, Shiyi Liang, Ruigang Niu, and Xing Wei. Adaptive capacity autoregressive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13574–13583, June 2026.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pp. 740–755, 2014.

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*, pp. 13424–13434, 2021.

Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, pp. 310–327, 2018.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Chon Hou Sio, Yu-Jen Ma, Hong-Han Shuai, Jun-Cheng Chen, and Wen-Huang Cheng. S^2SiamFC: Self-supervised fully convolutional siamese network for visual tracking. In *ACM Multimedia*, pp. 1948–1957, 2020.

Paul Voigtlaender, Jonathon Luiten, Philip H. S. Torr, and Bastian Leibe. Siam R-CNN: visual tracking by re-detection. In *CVPR*, pp. 6577–6587, 2020.

Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, pp. 2566–2576, 2019.

Fei Xie, Lei Chu, Jiahao Li, Yan Lu, and Chao Ma. Videotrack: Learning to track objects via video transformer. In *CVPR*, pp. 22826–22835, 2023.

Jinxia Xie, Bineng Zhong, Zhiyi Mo, Shengping Zhang, Liangtao Shi, Shuxiang Song, and Rongrong Ji. Autoregressive queries for adaptive tracking with spatio-temporal transformers. In *CVPR*, pp. 19300–19309, 2024.

Wei Xing, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *CVPR*, pp. 9697–9706, 2023.

Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, pp. 10428–10437, 2021.

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.

Di Yuan, Xiaojun Chang, Po-Yao Huang, Qiao Liu, and Zhenyu He. Self-supervised deep correlation tracking. *IEEE Transactions on Image Processing*, 30:976–985, 2020a.

Weihao Yuan, Michael Yu Wang, and Qifeng Chen. Self-supervised object tracking with cycle-consistent siamese networks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 10351–10358, 2020b.

Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, pp. 771–787, 2020.

Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. ODTrack: Online dense temporal token learning for visual tracking. In *AAAI*, volume 38, pp. 7588–7596, 2024.

Yaozong Zheng, Bineng Zhong, Qihua Liang, Ning Li, and Shuxiang Song. Decoupled spatio-temporal consistency learning for self-supervised tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 10635–10643, 2025.