CSPTRACK: CAUSAL STATE PREDICTION WITH RIDGE-AWARE TOKEN PRUNING FOR VISUAL TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Current template–search trackers often fail to sufficiently model the temporal evolution of target states, lack effective constraints for maintaining consistent target representations across frames, and densely process a large number of redundant search tokens, thereby limiting both tracking robustness and computational efficiency. To address these issues, we propose CSPTrack, a robust and efficient visual tracking framework with three complementary designs. First, the causal state prediction module leverages current search features and historical hidden states to explicitly predict short-term future-state representations, thereby modeling the temporal evolution of target states. Second, the instance-consistency objective brings the template and search representations of the same target closer while separating the representations of different target instances, enhancing both cross-frame representation consistency and target discrimination. Third, the ridge-aware token pruning strategy evaluates the importance of search tokens based on ridge-reconstruction errors, selectively retains informative search tokens, aggregates information from discarded tokens, and preserves all template tokens. Extensive experiments on six challenging benchmark datasets demonstrate the effectiveness and efficiency of CSPTrack. Notably, without token pruning, CSPTrack-B224 achieves a new state-of-the-art performance of 75.4% AUC on LaSOT under similar settings. *Even after reducing the number of search tokens by 30%, CSPTrack-B224 incurs only a marginal 0.2% AUC drop on LaSOT while improving tracking speed by 10%.* These results demonstrate that CSPTrack effectively reduces redundant computation while maintaining high tracking accuracy.

1 INTRODUCTION

Template–search feature interaction has become a dominant paradigm in single object tracking, achieving strong target localization and appearance discrimination (Chen et al., 2021; Ye et al., 2022; Cui et al., 2022; Chen et al., 2023; Kang et al., 2025). Nevertheless, existing trackers remain limited by insufficient temporal state modeling, unstable target identity under temporal fusion, and redundant token computation. These challenges, illustrated in Figure 1, restrict both tracking robustness and computational efficiency.

First, most template–search trackers emphasize spatial feature matching but lack explicit modeling of short-term target-state evolution (Cai et al., 2024; Zheng et al., 2024; Kang et al., 2025). Consequently, their current-frame representations may be insufficient under fast motion, occlusion, and substantial appearance variation. Second, temporal fusion propagates not only target information but also background responses and distractors. Without an instance-level constraint, the fused representation may exhibit target-identity ambiguity in complex scenes (Ma et al., 2024; Wang et al., 2021). Third, dense processing of search tokens introduces considerable redundant computation because many background tokens contribute little to localization (Rao et al., 2021; Xu et al., 2022; Wei et al., 2024; Ma et al., 2026; Wu et al., 2026a). Efficient token reduction must therefore preserve informative target-related features.

To address these issues, we propose **CSPTrack**, which integrates causal state prediction, target instance consistency, and ridge-aware token pruning within a unified tracking framework. The Causal State Prediction Module (CSPM) combines the current search representation with the historical hidden state produced by the neck to predict short-term future-state features, which are subsequently fused with the current representation. An instance-consistency objective regularizes the template and

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.

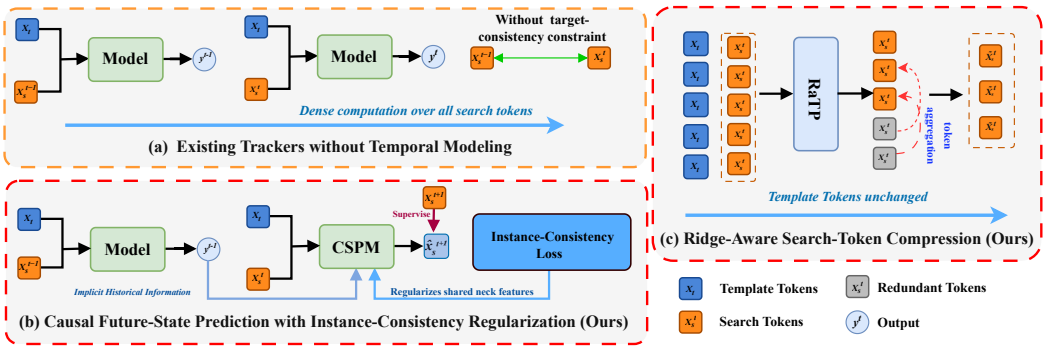

Figure 1: Comparison of conventional tracking paradigms with the proposed components of CSPTrack. (a) Conventional template–search trackers (Chen et al., 2021; Ye et al., 2022; Cui et al., 2022) process consecutive frames independently without explicit temporal state modeling or target-consistency constraints, while densely computing all search tokens. (b) CSPM incorporates historical information to predict short-term future-state representations under next-frame supervision, while the instance-consistency objective regularizes the shared neck features to improve target discrimination and prediction stability. (c) RaTP preserves all template tokens, retains informative search tokens according to their ridge-reconstruction errors, and aggregates redundant tokens into the retained representations to reduce subsequent computation.

search target features during training to improve target discrimination. In addition, the Ridge-Aware Token Pruning (RaTP) strategy estimates search-token redundancy through ridge reconstruction, retains informative tokens, and aggregates removed tokens without pruning the template tokens. The main contributions are summarized as follows:

- We propose a **Causal State Prediction Module** that jointly exploits current search features and historical hidden states to predict short-term future-state representations, improving target-state modeling under motion, occlusion, and appearance variation.
- We introduce a **instance-consistency constraint** in training-stage that aligns target representations across template and search frames, reducing feature confusion caused by complex backgrounds and visually similar distractors.
- We develop a **Ridge-Aware Token Pruning strategy** that evaluates search-token redundancy through ridge reconstruction and selectively preserves informative tokens, achieving a 10% increase in tracking speed while incurring only a marginal performance drop.

2 Related Work

2.1 Video Context Modeling for Tracking

Recent trackers have extended template–search interaction by incorporating temporal context. VideoTrack (Xie et al., 2023) learns spatiotemporal features from video clips, HIPTrack (Cai et al., 2024) uses historical prompts to preserve target cues, and ODTrack (Zheng et al., 2024) propagates online temporal tokens across frames. AQATrack (Xie et al., 2024) captures target changes through autoregressive queries, while TemTrack (Xie et al., 2025) models temporal relations among representative track tokens. SPMTrack (Cai et al., 2025) extends relation modeling to spatiotemporal context with a mixture of experts, and ARTrackV2 (Bai et al., 2024) jointly models the evolution of target location and appearance. MCITrack (Kang et al., 2025) maintains contextual hidden states throughout a video sequence. DreamTrack (Guo et al., 2025) further predicts plausible future variations and target trajectories from historical observations. Building on MCITrack's contextual propagation mechanism, CSPTrack introduces CSPM to predict a future-state feature representation from the current search representation and historical neck state, then uses it to enhance subsequent features.

2.2 Target Discrimination and Instance Consistency

Distinguishing the target from similar objects and background clutter is a central challenge in tracking. DaSiamRPN (Zhu et al., 2018) uses distractor-aware training and online adaptation, while DiMP (Bhat et al., 2019) learns a discriminative target model from target and background appearance. IAT (Wang et al., 2021) combines contrastive learning with negative-sample selection, and UVLTrack (Ma et al., 2024) uses contrastive learning to align visual and linguistic target representations. CSPTrack complements these approaches with a training-stage instance-consistency constraint that directly aligns template-target and search-target features of the same instance, strengthening their correspondence under background clutter and similar-object interference.

2.3 Efficient Tracking and Token Pruning

Token pruning reduces the computational cost of Transformer (Vaswani et al., 2017) models.DynamicViT (Rao et al., 2021) progressively removes tokens according to estimated importance, whereas Evo-ViT (Xu et al., 2022) reduces the processing of low-value tokens through slow–fast token evolution. In visual tracking, OSTrack (Ye et al., 2022) uses early candidate elimination, LiteTrack (Wei et al., 2024) combines asynchronous feature extraction with layer pruning, and UTPTrack (Wu et al., 2026a) jointly prunes template and search tokens. ETCTrack (Wu et al., 2026b) compresses tokens from historical templates to form a compact target representation. The proposed RaTP instead evaluates redundancy among search tokens through ridge reconstruction and aggregates removed tokens into retained representations. It leaves template tokens unpruned to preserve the target reference.

3 Method

3.1 Overall Framework of CSPTrack

We propose CSPTrack, an enhanced framework for efficient single-object tracking, whose overall architecture is illustrated in Fig. 2. CSPTrack follows the template–search interaction paradigm, in which the template region provides prior information about the target, while feature extraction, feature interaction, and target localization are performed within the search region of the current frame. Compared with existing trackers based on one-stream feature interaction or mixed-attention modeling (Ye et al., 2022; Cui et al., 2022), CSPTrack introduces a CSPM, an instance-consistency constraint, and a RaTP strategy to improve target-state modeling, preserve target identity, and enhance inference efficiency. To retain video-level contextual modeling, CSPTrack adopts the Contextual Information Fusion (CIF) module proposed in MCITrack (Kang et al., 2025). Its internal architecture and hidden-state update mechanism remain unchanged, enabling contextual information to propagate across video frames.

During feature interaction, CSPM combines the current search-region representation with the historical hidden state propagated by the neck to predict subsequent target-state features. The predicted features are fused with the current representation to improve adaptation to target motion, occlusion, and appearance variation. The instance-consistency constraint regularizes the correspondence between template and search target representations, while RaTP retains informative search tokens and reduces redundant feature interaction. Finally, the enhanced search representation is forwarded to the prediction head to generate the classification response and bounding-box regression result.

3.2 Causal State Prediction Module

We introduce a CSPM between contextual feature interaction and the prediction head. It combines the current search representation with the historical hidden state to predict a short-term future-state representation for subsequent feature enhancement. At time step t, let $\boldsymbol{X}_{\mathrm{s}}^{t} \in \mathbb{R}^{B\times N_{\mathrm{s}}\times C}$ denote the search tokens, where s identifies the search branch, B is the batch size, N_{s} is the number of search tokens, and C is the feature dimension. Given the hidden state $\boldsymbol{H}^{t-1}$ propagated by the neck from the preceding time step, the history-enhanced representation $\boldsymbol{Z}^{t}$ is computed as

$$\boldsymbol{Z}^{t} = \boldsymbol{X}_{\mathrm{s}}^{t} + \tanh(g)\,\phi_h\big(\mathrm{Pool}(\boldsymbol{H}^{t-1})\big)\,, \tag{1}$$

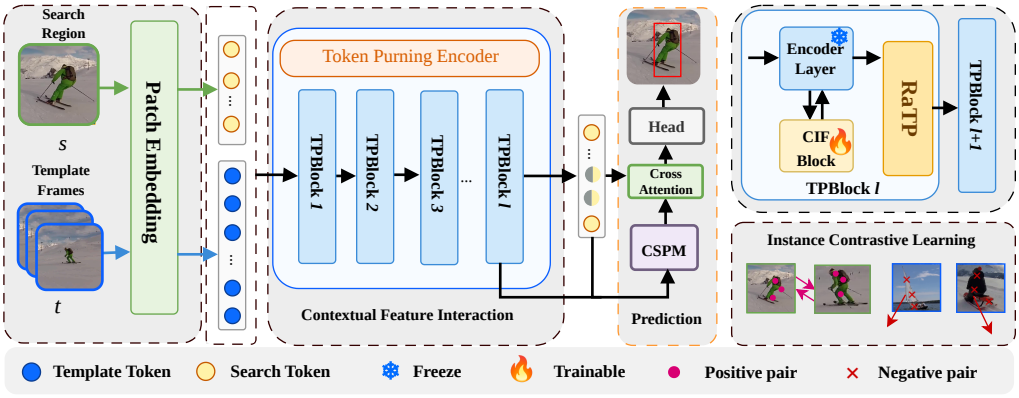

Figure 2: Overall architecture of CSPTrack. The template and search images are embedded into token sequences and processed by a series of TPBlocks. CIF propagates historical context during template–search interaction, while RaTP compresses redundant search tokens at selected interaction stages. CSPM predicts a future-state representation and integrates it with the current search representation before target localization. The target instance-consistency constraint is applied only during training.

where $\mathrm{Pool}(\cdot)$ aggregates the historical state, $\phi_h(\cdot)$ denotes its normalization and linear projection, the subscript h identifies the historical branch, and g is a learnable scalar gate bounded by $\tanh(\cdot)$. The historical term is omitted when no preceding state is available. CSPM further introduces two groups of learnable future-state queries, $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$, whose corresponding outputs are denoted by $\boldsymbol{F}_1^t$ and $\boldsymbol{F}_2^t$. After adding positional and segment embeddings, $\boldsymbol{E}_{\mathrm{pos}}$ and $\boldsymbol{E}_{\mathrm{seg}}$, the current and query tokens are jointly processed by stacked Temporal ViT blocks (Dosovitskiy et al., 2021):

$$[\boldsymbol{F}_1^t; \boldsymbol{F}_2^t] = \left[f_\theta\big([\boldsymbol{Z}^t; \boldsymbol{Q}_1; \boldsymbol{Q}_2] + \boldsymbol{E}_{\mathrm{pos}} + \boldsymbol{E}_{\mathrm{seg}}\big) \right]_{\boldsymbol{Q}_1, \boldsymbol{Q}_2}, \tag{2}$$

where $f_\theta(\cdot)$ is the temporal predictor parameterized by θ, $[\cdot;\cdot]$ denotes token concatenation, and the output subscript extracts the representations at the two query groups. The second prediction $\boldsymbol{F}_2^t$ is used for feature fusion and next-frame supervision during training. Specifically, the enhanced search representation $\boldsymbol{X}_{\mathrm{s,enh}}^t$, where 'enh' denotes feature enhancement, is obtained through cross-attention (Vaswani et al., 2017):

$$\boldsymbol{X}_{\mathrm{s,enh}}^t = \mathrm{LN}\big(\boldsymbol{X}_{\mathrm{s}}^t + \mathrm{MHA}\big(\boldsymbol{X}_{\mathrm{s}}^t, \boldsymbol{F}_2^t, \boldsymbol{F}_2^t\big)\big), \tag{3}$$

where $\mathrm{MHA}(\cdot)$ and $\mathrm{LN}(\cdot)$ denote multi-head attention and layer normalization, respectively. During training, $\boldsymbol{F}_2^t$ is supervised by the target heatmap of the next sampled search frame through an auxiliary causal state-prediction loss $\mathcal{L}_{\mathrm{csp}}$. During inference, CSPM uses only the current search tokens and preceding hidden state, without requiring future-frame information.

3.3 Target Instance Consistency Constraint

During training, we introduce a target instance-consistency constraint to align the representations of the same target across template and search frames while separating different target instances. Given a batch of B samples, let $\boldsymbol{X}_{\mathrm{tmp}} \in \mathbb{R}^{B\times N_{\mathrm{tmp}}\times C}$ denote the template tokens and $\boldsymbol{X}_{\mathrm{s}}^t \in \mathbb{R}^{B\times N_{\mathrm{s}}\times C}$ denote the tokens of the t-th sampled search frame, where N_{tmp} is the number of template tokens. Because both branches contain background tokens, directly averaging all tokens may weaken the target representation. We therefore construct Gaussian weights centered at the ground-truth target position to aggregate target-related features. For the i-th sample, $\boldsymbol{X}_{\mathrm{tmp},i,n}$ and $\boldsymbol{X}_{\mathrm{s},i,n}^t$ denote the n-th template and search tokens, while $\omega_{i,n}^{\mathrm{tmp}}$ and $\omega_{i,n}^{\mathrm{s},t}$ denote their corresponding normalized Gaussian weights. The template and search target representations are computed as

$$\boldsymbol{T}_i = \sum_{n=1}^{N_{\mathrm{tmp}}} \omega_{i,n}^{\mathrm{tmp}} \boldsymbol{X}_{\mathrm{tmp},i,n}, \qquad \boldsymbol{S}_i^t = \sum_{n=1}^{N_{\mathrm{s}}} \omega_{i,n}^{\mathrm{s},t} \boldsymbol{X}_{\mathrm{s},i,n}^t. \tag{4}$$

Here, $i \in \{1,\ldots,B\}$ indexes the training sample and n indexes the spatial token. The weights are normalized independently within each branch, such that $\sum_{n=1}^{N_{\mathrm{tmp}}} \omega_{i,n}^{\mathrm{tmp}} = 1$ and $\sum_{n=1}^{N_{\mathrm{s}}} \omega_{i,n}^{\mathrm{s},t} = 1$.

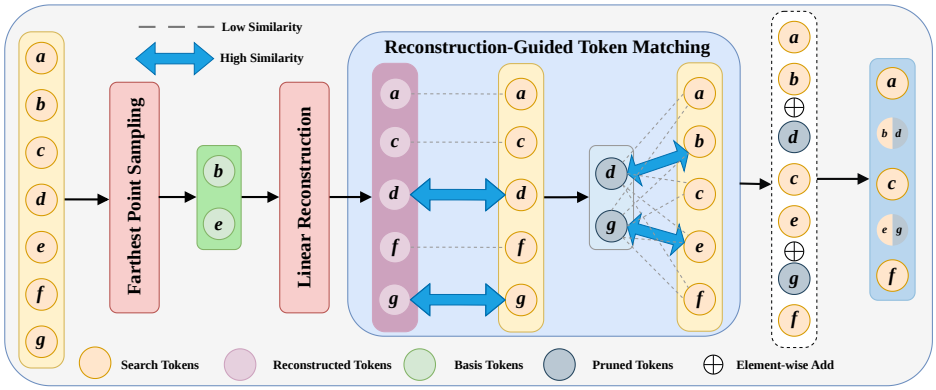

Figure 3: Overview of RaTP. Basis tokens are selected to reconstruct the search tokens, after which informative tokens are retained according to their reconstruction errors. Removed tokens are assigned to similar retained tokens for feature aggregation.

Consequently, $\boldsymbol{T}_i$ and $\boldsymbol{S}_i^t$ primarily encode the target regions in the template and search frames, respectively. Both representations are ℓ_2-normalized before their cosine similarity is evaluated. For each search representation $\boldsymbol{S}_i^t$, its corresponding template representation $\boldsymbol{T}_i$ forms a positive pair. The template representations $\boldsymbol{T}_j$ from the remaining samples, where $j \in \{1, \ldots, B\}$ and $j \neq i$, form negative pairs. We employ the InfoNCE loss $\mathcal{L}_{\text{ins}}$ (van den Oord et al., 2018). Let N_f denote the number of sampled search frames used for each training sample. In the following objective, $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, and $\tau > 0$ is the temperature parameter:

$$\mathcal{L}_{\text{ins}} = -\frac{1}{N_f B} \sum_{t=1}^{N_f} \sum_{i=1}^{B} \log \frac{\exp\left(\text{sim}\left(\boldsymbol{S}_i^t, \boldsymbol{T}_i\right)/\tau\right)}{\sum_{j=1}^{B} \exp\left(\text{sim}\left(\boldsymbol{S}_i^t, \boldsymbol{T}_j\right)/\tau\right)}. \tag{5}$$

The factor $1/(N_f B)$ averages the objective over all sampled search frames and batch samples. Minimizing this objective increases the similarity between representations of the same target while improving the discrimination between different target instances. The instance-consistency loss is incorporated into the overall training objective as an auxiliary term. The tracking loss, causal state-prediction loss, and their combination with the instance-consistency loss are detailed in Appendix S2. Since the constraint operates only on intermediate features during training, it introduces no additional inference branch.

3.4 Ridge-Aware Token Pruning

To reduce redundant feature interaction caused by background tokens, we propose RaTP. Motivated by approximation-error-guided token compression (Ma et al., 2026), RaTP adapts reconstruction-based redundancy estimation to template–search tracking. As illustrated in Fig. 3, it sequentially performs representative token selection, ridge reconstruction, reconstruction-error-guided retention, and similarity-based feature aggregation. Only search tokens are compressed, while all template tokens remain unchanged. For clarity, we describe RaTP for a single sample and omit the batch index. Let l index an interaction stage, N_l denote the number of search tokens entering that stage, and $\boldsymbol{X}_{\text{s}}^l \in \mathbb{R}^{N_l \times C}$ and $\boldsymbol{X}_{\text{tmp}}^l \in \mathbb{R}^{N_{\text{tmp}} \times C}$ denote its search- and template-token matrices, respectively. Given the retention ratio ρ_l, RaTP retains $K_l = \max\{M, \lfloor \rho_l N_l \rfloor\}$ search tokens, where M is the number of basis tokens, $\lfloor \cdot \rfloor$ is the floor operator, and $\max\{\cdot, \cdot\}$ selects the larger argument. The condition $K_l \geq M$ ensures that all selected basis tokens are retained. RaTP first applies Farthest Point Sampling (FPS) (Qi et al., 2017) to select representative tokens that provide dispersed coverage of the search-feature distribution. The selected basis-token index set is denoted by $\mathcal{I}_{\text{b}}^l = \text{FPS}(\boldsymbol{X}_{\text{s}}^l, M)$. The corresponding basis matrix is $\boldsymbol{B}^l = \boldsymbol{X}_{\text{s}}^l[\mathcal{I}_{\text{b}}^l] \in \mathbb{R}^{M \times C}$, where $\boldsymbol{X}_{\text{s}}^l[\mathcal{I}_{\text{b}}^l]$ denotes row selection using the indices in $\mathcal{I}_{\text{b}}^l$. Let $n \in \{1, \ldots, N_l\}$ index a search token and $\boldsymbol{x}_n^l \in \mathbb{R}^{1 \times C}$ denote its row feature vector. Using the selected basis matrix, this token is approximated through Ridge regression (Hoerl & Kennard, 1970). Let $\boldsymbol{w}_n^l \in \mathbb{R}^{1 \times M}$ denote its row vector of reconstruction coefficients,

$\widehat{\boldsymbol{x}}_n^l$ its reconstructed representation, and e_n^l its reconstruction error. They are computed as follows, with a detailed derivation provided in Appendix S1:

$$\begin{aligned} \boldsymbol{w}_n^l &= \boldsymbol{x}_n^l (\boldsymbol{B}^l)^\top \left[\boldsymbol{B}^l (\boldsymbol{B}^l)^\top + \lambda_{\mathrm{r}} \boldsymbol{I}_M\right]^{-1}, \\ \widehat{\boldsymbol{x}}_n^l &= \boldsymbol{w}_n^l \boldsymbol{B}^l, \qquad e_n^l = \left\|\boldsymbol{x}_n^l - \widehat{\boldsymbol{x}}_n^l\right\|_2 . \end{aligned} \tag{6}$$

Here, λ_{r} is the Ridge regularization coefficient; $\boldsymbol{I}_M \in \mathbb{R}^{M\times M}$ is the identity matrix, $(\cdot)^\top$ denotes matrix transpose, $(\cdot)^{-1}$ denotes matrix inverse, and $\|\cdot\|_2$ is the Euclidean norm. A small e_n^l indicates that the token can be adequately represented by the basis subspace and is therefore relatively redundant, whereas a large error indicates distinctive information and leads to a higher retention priority. Based on the reconstruction errors, RaTP preserves all basis tokens and selects an additional $K_l - M$ non-basis tokens with the largest errors. Let $\mathcal{I}_{\mathrm{s}}^l = \{1, \ldots, N_l\}$ denote the complete search-token index set and $\mathcal{I}_{\mathrm{k}}^l$ the retained-token index set. The retained set is

$$\mathcal{I}_{\mathrm{k}}^l = \mathcal{I}_{\mathrm{b}}^l \cup \mathrm{Top}_{K_l - M}\left\{e_n^l \,\middle|\, n \in \mathcal{I}_{\mathrm{s}}^l \setminus \mathcal{I}_{\mathrm{b}}^l\right\}, \tag{7}$$

where Top_{K_l-M} returns the indices of the $K_l - M$ largest reconstruction errors, and $\setminus$ denotes set difference. Thus, $\mathcal{I}_{\mathrm{k}}^l$ contains exactly K_l retained search-token indices. To preserve information from the removed tokens, RaTP defines the removed-token index set as $\mathcal{I}_{\mathrm{r}}^l = \mathcal{I}_{\mathrm{s}}^l \setminus \mathcal{I}_{\mathrm{k}}^l$. Let $r \in \mathcal{I}_{\mathrm{r}}^l$ index a removed token and $j \in \mathcal{I}_{\mathrm{k}}^l$ index a retained token. The assignment function $a^l(r)$ maps removed token r to its most similar retained token, $\mathcal{R}_j^l$ is the set of removed-token indices assigned to j, and $\boldsymbol{x}_{j,\mathrm{cmp}}^l$ is the resulting compressed representation. These quantities are computed as

$$\begin{aligned} a^l(r) &= \underset{j\in\mathcal{I}_{\mathrm{k}}^l}{\arg\max}\, \mathrm{sim}\left(\boldsymbol{x}_r^l, \boldsymbol{x}_j^l\right), \\ \mathcal{R}_j^l &= \left\{r \in \mathcal{I}_{\mathrm{r}}^l \,\middle|\, a^l(r) = j\right\}, \\ \boldsymbol{x}_{j,\mathrm{cmp}}^l &= \frac{\boldsymbol{x}_j^l + \sum\limits_{r\in\mathcal{R}_j^l} \boldsymbol{x}_r^l}{1 + \left|\mathcal{R}_j^l\right|}, \qquad j \in \mathcal{I}_{\mathrm{k}}^l. \end{aligned} \tag{8}$$

Here, $|\mathcal{R}_j^l|$ denotes the cardinality of $\mathcal{R}_j^l$. The averaging operation incorporates information from the removed tokens into their most relevant retained representations while preventing the feature magnitude from increasing with the number of assigned tokens. Finally, the aggregated tokens are restored according to the spatial order of their retained indices, producing the compressed search-token matrix $\boldsymbol{X}_{\mathrm{s,cmp}}^l \in \mathbb{R}^{K_l\times C}$. The compressed search tokens are then concatenated with the unchanged template tokens to form $\boldsymbol{X}_{\mathrm{in}}^{l+1} = [\boldsymbol{X}_{\mathrm{s,cmp}}^l; \boldsymbol{X}_{\mathrm{tmp}}^l] \in \mathbb{R}^{(K_l+N_{\mathrm{tmp}})\times C}$, where $\boldsymbol{X}_{\mathrm{in}}^{l+1}$ denotes the input to the next interaction stage $l+1$. By reducing the number of search tokens from N_l to K_l, RaTP effectively reduces the computation required by subsequent feature interaction and improves inference efficiency.

4 Experiments

4.1 Implementation Details

Training. We use a frozen Fast-iTPN-B (Tian et al., 2023) encoder as a pretrained checkpoint. The remaining modules are trained on LaSOT, GOT-10k, COCO, TrackingNet, and VastTrack with equal sampling ratios. Each sample contains five 112×112 templates and four 224×224 search frames, with 60,000 samples generated per epoch. CSPM contains four Temporal ViT blocks with eight attention heads. The temporal prediction and instance-consistency losses are weighted by 0.1 and 0.02, respectively. We use AdamW (Loshchilov & Hutter, 2019) with a batch size of 10, an initial learning rate of 4×10^{-4}, and a weight decay of 10^{-4}. Training lasts for 300 epochs, with the learning rate reduced by a factor of 10 after 240 epochs. For GOT-10k, we separately train the model using only its official training split.

Table 1: Comparison with state-of-the-art trackers on GOT-10k, LaSOT, LaSOT$_{\text{ext}}$, and TrackingNet. All results are reported as percentages. The best and second-best results are highlighted in bold and underlined, respectively. * denotes evaluation under the GOT-10k one-shot protocol, where the model is trained only on the official GOT-10k training split.

Method	Source	GOT-10k*			LaSOT			LaSOT$_{\text{ext}}$			TrackingNet		
		AO	SR$_{0.5}$	SR$_{0.75}$	AUC	P$_{\text{norm}}$	P	AUC	P$_{\text{norm}}$	P	AUC	P$_{\text{norm}}$	P
CSPTrack-B224	Ours	<u>78.9</u>	<u>88.6</u>	**79.4**	**75.4**	**85.7**	**83.3**	**54.6**	<u>65.8</u>	<u>62.1</u>	**86.6**	**91.0**	**86.4**
CSPTrack-B224$_{\rho=0.7}$	Ours	78.5	88.2	<u>79.2</u>	75.2	85.5	<u>83.1</u>	54.4	65.6	61.9	86.4	<u>90.8</u>	86.2
ETCTrack-B224 Wu et al. (2026b)	CVPR26	**79.2**	**90.2**	78.6	74.9	85.1	82.7	**54.6**	**65.9**	**62.4**	86.0	90.6	85.6
UTPTrack-S224 Wu et al. (2026a)	CVPR26	77.3	87.1	77.7	72.6	82.8	79.8	53.6	64.8	61.4	85.2	90.0	84.7
MCITrack-B224 Kang et al. (2025)	AAAI25	77.9	87.5	78.8	<u>75.3</u>	<u>85.6</u>	**83.3**	52.0	62.5	59.0	86.4	**91.0**	<u>86.3</u>
DreamTrack-B256 Guo et al. (2025)	CVPR25	77.5	87.1	74.2	73.8	83.4	80.6	53.1	64.1	59.8	85.8	90.0	85.3
ARPTrack-B256 Liang et al. (2025)	CVPR25	77.7	87.3	74.3	72.6	81.4	78.5	52.0	62.9	58.7	85.5	90.0	85.3
TemTrack-B256 Xie et al. (2025)	AAAI25	74.9	84.8	71.7	72.0	82.1	79.1	52.4	63.3	60.2	84.3	88.8	83.5
ARTrackV2-B256 Bai et al. (2024)	CVPR24	75.9	85.4	72.7	71.6	80.2	77.2	50.8	61.9	57.7	84.9	89.3	84.5
VideoTrack Xie et al. (2023)	CVPR23	72.9	81.9	69.8	70.2	–	76.4	–	–	–	83.8	88.7	83.1
SPMTrack-B384 Cai et al. (2025)	CVPR25	76.5	85.9	76.3	74.9	84.0	81.7	–	–	–	86.1	90.2	85.6
DreamTrack-B384 Guo et al. (2025)	CVPR25	78.3	87.9	76.6	75.0	84.2	81.7	<u>54.5</u>	65.3	61.1	<u>86.5</u>	90.6	85.9
HIPTrack-B384 Cai et al. (2024)	CVPR24	77.4	88.0	74.5	72.7	82.9	79.5	53.0	64.3	60.6	84.5	89.1	83.8
ODTrack-B384 Zheng et al. (2024)	AAAI24	77.0	87.9	75.1	73.2	83.2	80.6	52.4	63.9	60.1	85.1	90.1	84.9

Inference. CSPTrack uses one 224×224 search crop and five 112×112 templates. For the efficient CSPTrack-B224$_{\rho=0.7}$ variant, RaTP retains 70% of the search tokens while preserving all template tokens. The unpruned version disables RaTP with all other settings unchanged. The initial template remains fixed, while the other templates are updated according to the tracking confidence. A Hanning window is applied before bounding-box decoding. On the NFS dataset, the pruned CSPTrack-B224$_{\rho=0.7}$ achieves an inference speed of 59 FPS on a single NVIDIA GeForce RTX 4090 GPU.

4.2 Comparison with State-of-the-Art Trackers

We evaluate CSPTrack-B224 on six benchmarks: GOT-10k, LaSOT, LaSOT$_{\text{ext}}$, TrackingNet, NFS, and OTB100. The main comparison is reported in Table 1, while the additional benchmark results are summarized in Table 2.

GOT-10k (Huang et al., 2021) is a large-scale tracking benchmark containing more than 10,000 video segments and covering 563 object classes and 87 motion patterns. It follows a one-shot protocol that requires the object classes in the training and test sets to be disjoint. Accordingly, CSPTrack is trained exclusively on the official GOT-10k training split. As reported in Table 1, the unpruned CSPTrack-B224 obtains 78.9% AO, 88.6% SR$_{0.5}$, and 79.4% SR$_{0.75}$, outperforming the strong DreamTrack-B384 by 0.6%, 0.7%, and 2.8%, respectively.

LaSOT and LaSOT$_{\text{ext}}$ (Fan et al., 2019; 2021) are large-scale long-term tracking benchmarks for evaluating tracking robustness and category generalization. We report AUC, normalized precision P$_{\text{norm}}$, and precision P in Table 1. CSPTrack-B224 achieves 75.4% AUC on LaSOT, outperforming DreamTrack-B384 at 75.0% and SPMTrack-B384 at 74.9%. It also obtains 54.6% AUC on LaSOT$_{\text{ext}}$, slightly exceeding DreamTrack-B384 at 54.5%. As shown in Figure 4, CSPTrack-B224 performs consistently well across the 14 challenge attributes, particularly under camera motion, deformation, occlusion, scale variation, and fast motion. The supplementary attribute-wise analysis reports detailed margins against competing trackers, further demonstrating robust performance across diverse tracking challenges.

TrackingNet (Müller et al., 2018) is a large-scale tracking dataset with more than 30,000 training sequences and 511 test sequences collected from unconstrained online videos. It covers challenging factors such as background clutter and low resolution. The AUC, normalized precision, and precision results are reported in Table 1. CSPTrack-B224 achieves 86.6% AUC, outperforming DreamTrack-B384 at 86.5% and SPMTrack-B384 at 86.1%, while attaining 86.4% precision.

OTB100 and NFS (Wu et al., 2015; Kiani Galoogahi et al., 2017) are two widely used tracking benchmarks. OTB100 consists of 100 sequences annotated with 11 challenge attributes, whereas NFS contains 100 high-frame-rate videos and places particular emphasis on fast target motion. We report AUC on both datasets in Table 2. CSPTrack-B224 achieves 71.5% AUC on NFS, exceeding

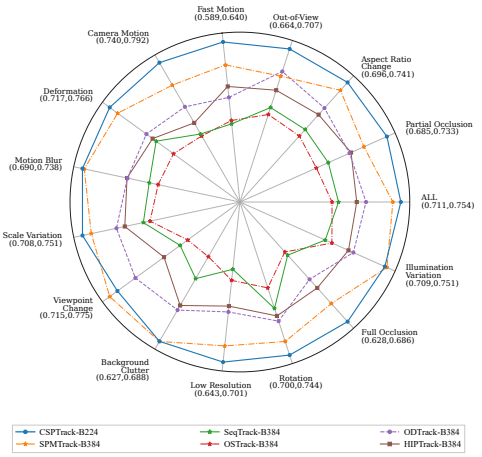

Figure 4: The performance of our method compared with other state-of-the-art trackers in terms of AUC across various scenarios in the LaSOT test split.

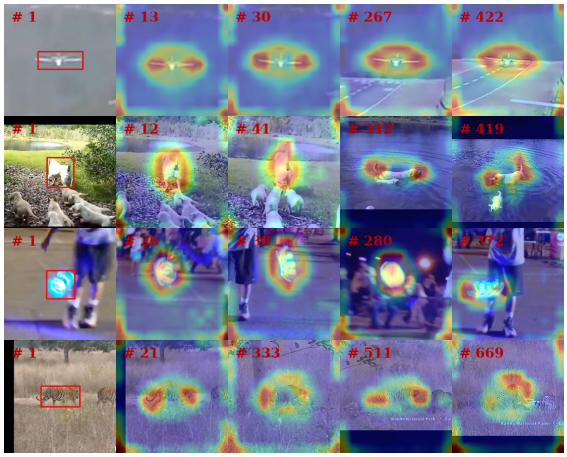

Figure 5: Visualization of CSPM-enhanced features on four LaSOT sequences. The first column shows the template frames, while the remaining columns show selected search frames.

Table 2: Comparison with state-of-the-art trackers on NFS and OTB100 in terms of AUC. The best and second-best results are highlighted using boldface and underlining, respectively.

	DreamTrack-B384	SPMTrack-B384	HIPTrack-B384	ODTrack-B384	LoRAT-L378	SeqTrack-B384	ETCTrack-B224	**CSPTrack-B224**$_{\rho=0.7}$	**CSPTrack-B224**
NFS	66.1	67.4	68.1	–	66.7	66.7	<u>71.3</u>	<u>71.3</u>	**71.5**
OTB100	72.0	72.7	71.0	72.3	–	–	**73.3**	<u>73.1</u>	**73.3**

Table 3: Component ablation on LaSOT. IC denotes instance consistency. Rel. Perf. is computed relative to the baseline, and the value in parentheses denotes the percentage-point difference.

Variant	CSPM	IC	RaTP	AUC	Rel. Perf. (%)
Baseline	–	–	–	74.0	100.0
w/o CSPM	–	✓	✓	74.6	100.8 (+0.8)
w/o IC	✓	–	✓	74.8	101.1 (+1.1)
w/o RaTP	✓	✓	–	**75.4**	**101.9** (+1.9)
Full	✓	✓	✓	<u>75.2</u>	<u>101.6</u> (+1.6)

Table 4: Search-token retention-ratio ablation on LaSOT. Rel. Perf. is computed relative to the unpruned baseline, and the value in parentheses denotes the percentage-point difference.

Keep Ratio ρ	AUC	P_{norm}	P	MACs (G)	Rel. Perf. (%)
1.0	**75.4**	**85.4**	**83.3**	48.6	**100.0**
0.7	<u>75.2</u>	<u>85.2</u>	<u>83.1</u>	46.8	<u>99.7</u> (-0.3)
0.6	75.0	85.0	82.9	<u>46.5</u>	99.5 (-0.5)
0.5	74.6	84.6	82.4	**46.2**	98.9 (-1.1)

Table 5: Search-frame ablation on LaSOT. Δ denotes the AUC change relative to the four-frame baseline.

Search Frames	AUC	P_{norm}	P	Δ
1 frame	74.3	84.1	81.8	**-0.9**
2 frames	74.9	84.9	82.8	**-0.3**
3 frames	<u>75.1</u>	<u>85.1</u>	<u>83.0</u>	**-0.1**
4 frames	**75.2**	**85.2**	**83.1**	–

Table 6: Template-number ablation on LaSOT. Δ denotes the AUC change relative to the five-template baseline.

Templates	AUC	P_{norm}	P	Δ
3 templates	74.7	84.6	82.4	**-0.5**
4 templates	<u>75.0</u>	85.0	82.8	**-0.2**
5 templates	**75.2**	**85.2**	**83.1**	–
6 templates	74.9	<u>85.1</u>	<u>83.0</u>	**-0.3**

Table 7: Basis-token selection on NFS ($M = 10, \rho = 0.7$). Δ is the AUC change relative to FPS.

Selection Strategy	AUC	FPS	Δ
Random	70.7	**60.3**	**-0.8**
K-means Medoid	<u>71.2</u>	55.6	**-0.3**
Spatial Uniform	70.9	<u>59.7</u>	**-0.6**
FPS (Ours)	**71.5**	59.0	–

ETCTrack-B224 71.3% by 0.2%. On OTB100, it obtains 73.3% AUC, tying with ETCTrack-B224 for the best result and outperforming SPMTrack-B384 72.7% by 0.6%.

4.3 Ablation Study and Analysis

We conduct five groups of ablation experiments using CSPTrack-B224 with a frozen backbone. The basis-token selection strategies are compared on NFS, while the remaining ablation experiments are conducted on LaSOT. Unless otherwise specified, each training sample contains four search frames and five template frames, five templates are used during inference, and the search-token retention ratio is set to $\rho = 0.7$.

Effect of individual components. Table 3 evaluates the contributions of CSPM, the instance-consistency constraint (IC), and RaTP. The baseline without these three components obtains 74.0% AUC, whereas the full model achieves 75.2% AUC, demonstrating an overall improvement of 1.2%. Removing CSPM decreases the AUC from 75.2% to 74.6%, confirming that explicit short-term state prediction improves adaptation to temporal motion and appearance changes. Removing IC reduces the AUC to 74.8%, showing that aligning target representations across template and search frames enhances target discrimination. The model without RaTP obtains 75.4% AUC, while the full model with RaTP achieves 75.2% AUC. Therefore, reducing the number of search tokens by 30% causes only a 0.2% performance drop, indicating that RaTP primarily removes redundant tokens while preserving target-relevant information.

Effect of the search-token retention ratio. As shown in Table 4, the unpruned model ($\rho = 1.0$) achieves 75.4% AUC with 48.6 G MACs. Retaining 70% of the search tokens reduces the computation to 46.8 G MACs while maintaining 75.2% AUC, corresponding to 99.7% of the unpruned performance. More aggressive pruning gradually reduces tracking accuracy: $\rho = 0.6$ and $\rho = 0.5$ produce 75.0% and 74.6% AUC, respectively. These results demonstrate that $\rho = 0.7$ provides a favorable balance between tracking accuracy and computational cost.

Number of sampled search frames. Table 5 investigates the influence of temporal context during training. Increasing the number of search frames from one to two improves the AUC from 74.3% to 74.9%, while using three frames further increases it to 75.1%. The best result of 75.2% AUC is obtained with four search frames, representing a total improvement of 0.9% over the single-frame setting. The consistent improvements in normalized precision and precision further demonstrate that multiple search frames provide useful temporal cues. Nevertheless, the gradually diminishing gains indicate that most of the temporal benefit is obtained from the first few search frames.

Number of templates. As reported in Table 6, increasing the number of templates from three to four improves the AUC from 74.7% to 75.0%, while five templates achieve the best result of 75.2% AUC. Increasing the number to six slightly decreases the AUC to 74.9%. This result suggests that additional templates initially provide complementary target-appearance information, whereas excessive templates may introduce redundant or outdated target cues. We therefore employ five templates as the default inference setting.

Effect of basis-token selection. Table 7 compares four basis-token selection strategies in RaTP on NFS under identical settings. Random and spatial uniform sampling select basis tokens without considering their feature distributions. The K-means Medoid strategy is based on K-means clustering (Lloyd, 1982) and selects the token nearest to each feature-cluster center. FPS (Qi et al., 2017) iteratively selects the token farthest from the current basis set, thereby encouraging diverse coverage of the feature space. FPS achieves the best AUC of 71.5%, outperforming K-means Medoid by 0.3 percentage points, which demonstrates the benefit of diverse feature-space coverage for token-redundancy estimation.

Visualization. Figure 5 visualizes the feature enhancement produced by CSPM on four LaSOT sequences. The first column shows the template frames, while the remaining columns show selected search frames, where the highlight regions indicate stronger feature changes. Despite target motion, appearance variation, and background interference, the responses remain concentrated on the target. This demonstrates that CSPM strengthens target-relevant features for robust localization.

5 Conclusion

In this work, we introduced CSPTrack, a purely visual tracking framework that improves temporal target modeling and reduces redundant token computation. Our framework consists of three key designs: CSPM, which exploits current search features and historical states to predict short-term future target representations; IC, which enhances target discrimination across template and search frames; and RaTP, which retains informative search tokens and aggregates redundant ones according to their reconstruction characteristics. Extensive experiments on six challenging benchmarks demonstrate that CSPTrack achieves competitive tracking performance while maintaining computational efficiency. We believe these results indicate that combining explicit target-state prediction with reconstruction-guided token compression is a promising direction for developing robust and efficient visual trackers.

AI USE STATEMENT

During the preparation of this work, we used generative AI tools to provide feedback on research methodology, assist with the implementation of CSPTrack, draft parts of the manuscript, and improve language readability. All AI-assisted content was carefully reviewed, verified, and revised by the authors. The authors take full responsibility for the methodology, experiments, results, and content presented in this work.

REPRODUCIBILITY STATEMENT

The overall architecture of CSPTrack and its principal components, including Causal State Prediction Modeling (CSPM), the instance-consistency objective, and Ridge-Aware Token Pruning (RaTP), are described in Section 3.1 and the subsequent method subsections. The training objectives, implementation details, data-processing procedures, hyperparameters, and training protocol are provided in Section 4.1 and the supplementary material. All experiments are conducted on publicly available tracking datasets and evaluated on public benchmarks following their standard evaluation protocols. The source code, trained models, and configuration files will be released upon acceptance.

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.

Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pp. 6182–6191, 2019.

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

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

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. In *CVPR*, pp. 14572–14581, 2023.

Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. MixFormer: End-to-end tracking with iterative mixed attention. In *CVPR*, pp. 13608–13618, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

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

Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, Yong Xu, Chunyuan Liao, Lin Yuan, and Haibin Ling. LaSOT: A high-quality large-scale single object tracking benchmark. *IJCV*, 129(2):439–461, 2021.

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

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 43(5):1562–1577, 2021.

Ben Kang, Xin Chen, Simiao Lai, Yang Liu, Yi Liu, and Dong Wang. Exploring enhanced contextual information for video-level object tracking. In *AAAI*, pp. 4194–4202, 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*, pp. 1125–1134, 2017.

Shiyi Liang, Yifan Bai, Yihong Gong, and Xing Wei. Autoregressive sequential pretraining for visual tracking. In *CVPR*, pp. 7254–7264, 2025.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pp. 2980–2988, 2017.

Stuart Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–137, 1982.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Qiankun Ma, Ziyao Zhang, Haofei Wang, Zhen Song, Jie Chen, and Hairong Zheng. ApET: Approximation-error guided token compression for efficient VLMs. In *CVPR*, pp. 26306–26316, 2026.

Yinchao Ma, Yuyang Tang, Wenfei Yang, Tianzhu Zhang, Jinpeng Zhang, and Mengxue Kang. Unifying visual and vision-language tracking via contrastive learning. In *AAAI*, pp. 4107–4116, 2024.

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

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pp. 5099–5108, 2017.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, pp. 13937–13949, 2021.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pp. 658–666, 2019.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

Mengmeng Wang, Xiaoqian Yang, and Yong Liu. Explicitly modeling the discriminability for instance-aware visual object tracking. *arXiv preprint arXiv:2110.15030*, 2021.

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*, pp. 4968–4975, 2024.

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

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

Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 37(9):1834–1848, 2015.

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.

Jinxia Xie, Bineng Zhong, Qihua Liang, Ning Li, Zhiyi Mo, and Shuxiang Song. Robust tracking via mamba-based context-aware token learning. In *AAAI*, volume 39, pp. 8727–8735, 2025. doi: 10.1609/aaai.v39i8.32943.

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-ViT: Slow-fast token evolution for dynamic vision transformer. In *AAAI*, pp. 2964–2972, 2022.

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.

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

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, pp. 101–117, 2018.

S SUPPLEMENTARY MATERIAL

Unless otherwise stated, the notation in this supplementary material follows the definitions in the main paper. In this supplementary material, we provide further methodological derivations and experimental analyses to complement the main paper. In Section S1, we first present the detailed derivation of the closed-form Ridge-regression solution used in Eq. 6, clarifying how RaTP reconstructs search tokens from the selected basis-token subspace. In Section S2, we introduce the complete training objective of CSPTrack and describe how the original tracking losses, the causal state-prediction loss, and the instance-consistency loss are combined during optimization. In Section S3, we report a comprehensive attribute-wise comparison covering all 14 LaSOT challenge attributes to provide a detailed evaluation of the robustness of CSPTrack. Finally, in Section S4, we present three groups of qualitative visualizations on representative LaSOT sequences to illustrate the behavior of CSPTrack under different challenging conditions.

S1 DETAILED DERIVATION OF RIDGE RECONSTRUCTION

RaTP estimates the redundancy of each search token according to how accurately it can be reconstructed from a compact set of representative basis tokens. This section provides the detailed derivation of the closed-form Ridge-regression solution used in Eq. 6. At the l-th interaction stage, let $\boldsymbol{x}_n^l \in \mathbb{R}^{1\times C}$ denote the feature of the n-th search token. After selecting M representative basis tokens, we organize their features as the rows of the basis matrix $\boldsymbol{B}^l \in \mathbb{R}^{M\times C}$. The objective is to approximate $\boldsymbol{x}_n^l$ by a linear combination of these basis tokens. Let $\boldsymbol{w}_n^l \in \mathbb{R}^{1\times M}$ denote the corresponding reconstruction coefficients. The reconstructed token is therefore expressed as $\widehat{\boldsymbol{x}}_n^l = \boldsymbol{w}_n^l \boldsymbol{B}^l$. Direct least-squares estimation may become unstable when the basis tokens contain correlated or nearly redundant features. We therefore introduce an ℓ_2-regularization term and determine the coefficients by minimizing the following Ridge-regression objective:

$$\mathcal{J}(\boldsymbol{w}_n^l) = \frac{1}{2}\left\|\boldsymbol{x}_n^l - \boldsymbol{w}_n^l \boldsymbol{B}^l\right\|_2^2 + \frac{\lambda_{\mathrm{r}}}{2}\left\|\boldsymbol{w}_n^l\right\|_2^2, \tag{S1}$$

where $\mathcal{J}(\boldsymbol{w}_n^l)$ denotes the regularized reconstruction objective for the n-th search token and $\lambda_{\mathrm{r}} > 0$ is the Ridge regularization coefficient. The first term measures the reconstruction error, while the second term constrains the magnitude of the reconstruction coefficients and improves numerical stability. To obtain the closed-form solution, we first expand Eq. S1 as

$$\begin{aligned}\mathcal{J}(\boldsymbol{w}_n^l) = &\frac{1}{2}\boldsymbol{x}_n^l(\boldsymbol{x}_n^l)^\top - \boldsymbol{w}_n^l \boldsymbol{B}^l(\boldsymbol{x}_n^l)^\top \\ &+ \frac{1}{2}\boldsymbol{w}_n^l \boldsymbol{B}^l(\boldsymbol{B}^l)^\top(\boldsymbol{w}_n^l)^\top + \frac{\lambda_{\mathrm{r}}}{2}\boldsymbol{w}_n^l(\boldsymbol{w}_n^l)^\top.\end{aligned} \tag{S2}$$

Taking the derivative with respect to $\boldsymbol{w}_n^l$ gives

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{w}_n^l} = \left(\boldsymbol{w}_n^l \boldsymbol{B}^l - \boldsymbol{x}_n^l\right)(\boldsymbol{B}^l)^\top + \lambda_{\mathrm{r}}\boldsymbol{w}_n^l. \tag{S3}$$

Setting the derivative to zero yields the normal equation

$$\boldsymbol{w}_n^l\left[\boldsymbol{B}^l(\boldsymbol{B}^l)^\top + \lambda_{\mathrm{r}}\boldsymbol{I}_M\right] = \boldsymbol{x}_n^l(\boldsymbol{B}^l)^\top, \tag{S4}$$

where $\boldsymbol{I}_M \in \mathbb{R}^{M\times M}$ is the identity matrix. Since $\boldsymbol{B}^l(\boldsymbol{B}^l)^\top$ is positive semidefinite and $\lambda_{\mathrm{r}} > 0$, the matrix $\boldsymbol{B}^l(\boldsymbol{B}^l)^\top + \lambda_{\mathrm{r}}\boldsymbol{I}_M$ is positive definite and therefore invertible. Right-multiplying both sides of Eq. S4 by its inverse gives

$$\boldsymbol{w}_n^l = \boldsymbol{x}_n^l(\boldsymbol{B}^l)^\top\left[\boldsymbol{B}^l(\boldsymbol{B}^l)^\top + \lambda_{\mathrm{r}}\boldsymbol{I}_M\right]^{-1}. \tag{S5}$$

This is the coefficient solution reported in Eq. 6 of the main paper. Substituting Eq. S5 into the reconstruction expression produces

$$\widehat{\boldsymbol{x}}_n^l = \boldsymbol{x}_n^l(\boldsymbol{B}^l)^\top\left[\boldsymbol{B}^l(\boldsymbol{B}^l)^\top + \lambda_{\mathrm{r}}\boldsymbol{I}_M\right]^{-1}\boldsymbol{B}^l. \tag{S6}$$

For further interpretation, we define the regularized projection matrix

$$\boldsymbol{P}_{\boldsymbol{B}}^{l} = (\boldsymbol{B}^{l})^{\top}\left[\boldsymbol{B}^{l}(\boldsymbol{B}^{l})^{\top} + \lambda_{\mathrm{r}}\boldsymbol{I}_{M}\right]^{-1}\boldsymbol{B}^{l}, \tag{S7}$$

where $\boldsymbol{P}_{\boldsymbol{B}}^{l} \in \mathbb{R}^{C\times C}$ denotes the regularized projection matrix induced by the basis matrix $\boldsymbol{B}^{l}$. It follows that $\widehat{\boldsymbol{x}}_{n}^{l} = \boldsymbol{x}_{n}^{l}\boldsymbol{P}_{\boldsymbol{B}}^{l}$. The reconstruction error can consequently be written as

$$\begin{aligned} e_{n}^{l} &= \left\|\boldsymbol{x}_{n}^{l} - \widehat{\boldsymbol{x}}_{n}^{l}\right\|_{2} \\ &= \left\|\boldsymbol{x}_{n}^{l}\left(\boldsymbol{I}_{C} - \boldsymbol{P}_{\boldsymbol{B}}^{l}\right)\right\|_{2}, \end{aligned} \tag{S8}$$

where $\boldsymbol{I}_{C} \in \mathbb{R}^{C\times C}$ denotes the identity matrix in the token-feature space. Equation S8 measures the feature component that cannot be sufficiently represented by the selected basis-token subspace. A small reconstruction error indicates that the token is well explained by the basis tokens and is therefore likely to contain redundant information. Conversely, a large error indicates that the token contains distinctive information outside the dominant basis subspace. RaTP consequently assigns higher retention priority to tokens with larger reconstruction errors, while tokens with smaller errors can be removed and subsequently aggregated into their most similar retained tokens. The formulation in Eq. S5 requires the inversion of an $M \times M$ matrix rather than a $C \times C$ matrix. Since the number of basis tokens is substantially smaller than the feature dimension in our implementation, i.e., $M \ll C$, this formulation provides an efficient way to evaluate token redundancy. Moreover, the same inverse matrix is shared by all search tokens within an interaction stage, and therefore needs to be computed only once for each sample and stage.

S2 Overall Training Objective

The training loss of CSPTrack consists of three components: the standard tracking loss, the causal state-prediction loss, and the target instance-consistency objective. The overall loss is defined as

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{trk}} + \lambda_{\mathrm{csp}}\mathcal{L}_{\mathrm{csp}} + \lambda_{\mathrm{ins}}\mathcal{L}_{\mathrm{ins}}, \tag{S9}$$

where $\mathcal{L}_{\mathrm{trk}}$ supervises target localization, $\mathcal{L}_{\mathrm{csp}}$ constrains the current- and future-state responses produced by CSPM, and $\mathcal{L}_{\mathrm{ins}}$ improves the consistency and discrimination of target representations across template and search frames. The scalars λ_{csp} and λ_{ins} are their respective loss weights. In our implementation, we set $\lambda_{\mathrm{csp}} = 0.1$ and $\lambda_{\mathrm{ins}} = 0.02$.

Tracking objective. Let N_f denote the number of sampled search frames and $t \in \{1, \ldots, N_f\}$ their temporal index. For frame t, the prediction head produces a classification response map $\widehat{\boldsymbol{G}}^{t}$ and a normalized bounding-box prediction $\widehat{\boldsymbol{b}}^{t}$. The symbols $\boldsymbol{G}^{t}$ and $\boldsymbol{b}^{t}$ denote the Gaussian target heatmap and normalized ground-truth box, respectively. The hat therefore distinguishes a prediction from its supervision target. The frame-level tracking loss is formulated as

$$\mathcal{L}_{\mathrm{trk}}^{t} = \lambda_{\mathrm{giou}}\mathcal{L}_{\mathrm{giou}}^{t} + \lambda_{\ell_1}\mathcal{L}_{\ell_1}^{t} + \lambda_{\mathrm{focal}}\mathcal{L}_{\mathrm{focal}}^{t}, \tag{S10}$$

where λ_{giou}, λ_{ℓ_1}, and λ_{focal} are the weights of the generalized-IoU, coordinate-regression, and classification losses, respectively, and

$$\mathcal{L}_{\mathrm{giou}}^{t} = 1 - \mathrm{GIoU}\left(\widehat{\boldsymbol{b}}^{t}, \boldsymbol{b}^{t}\right), \qquad \mathcal{L}_{\ell_1}^{t} = \left\|\widehat{\boldsymbol{b}}^{t} - \boldsymbol{b}^{t}\right\|_{1}. \tag{S11}$$

Here, $\mathrm{GIoU}(\cdot,\cdot)$ denotes generalized intersection over union (Rezatofighi et al., 2019). Both predicted and ground-truth boxes are converted to the corner-coordinate format before evaluating these two regression losses. The classification response is supervised using a CenterNet-style Focal Loss (Lin et al., 2017; Zhou et al., 2019). Let p index a spatial location, and let $\widehat{G}_{p}^{t} \in (0,1)$ and $G_{p}^{t} \in [0,1]$ denote the predicted response and Gaussian target value at that location, respectively. The classification loss is

$$\begin{aligned} \mathcal{L}_{\mathrm{focal}}^{t} = -\frac{1}{N_{+}^{t}}\Bigg[&\sum_{p:G_{p}^{t}=1}\left(1-\widehat{G}_{p}^{t}\right)^{\alpha_{\mathrm{f}}}\log\widehat{G}_{p}^{t} \\ &+ \sum_{p:G_{p}^{t}<1}\left(1-G_{p}^{t}\right)^{\beta_{\mathrm{f}}}\left(\widehat{G}_{p}^{t}\right)^{\alpha_{\mathrm{f}}}\log\left(1-\widehat{G}_{p}^{t}\right)\Bigg], \end{aligned} \tag{S12}$$

where N_+^t is the number of positive locations and α_{f} and β_{f} are the positive and negative focusing exponents, respectively. Following the implementation of the prediction head, we use $\alpha_{\mathrm{f}} = 2$ and $\beta_{\mathrm{f}} = 4$. The complete tracking objective is accumulated over the N_f sampled search frames:

$$\mathcal{L}_{\mathrm{trk}} = \sum_{t=1}^{N_f} \mathcal{L}_{\mathrm{trk}}^t. \tag{S13}$$

The corresponding loss weights are $\lambda_{\mathrm{giou}} = 2$, $\lambda_{\ell_1} = 5$, and $\lambda_{\mathrm{focal}} = 1$.

Causal state-prediction objective. CSPM produces the current search representation $\boldsymbol{X}_{\mathrm{s}}^t$ and the predicted future-state representation $\boldsymbol{F}_2^t$. A shared 1×1 convolutional head projects these two representations into the current-state logits $\boldsymbol{H}_{\mathrm{cur}}^t$ and the future-state logits $\boldsymbol{H}_{\mathrm{pred}}^t$, respectively. The current-state output is supervised by $\boldsymbol{G}^t$, while the predicted future output is supervised by the Gaussian heatmap $\boldsymbol{G}^{t+1}$ of the next sampled search frame. For a predicted logit map $\boldsymbol{H}$ and a Gaussian target map $\boldsymbol{G}$, the implementation employs a foreground-aware weighted L1 loss:

$$\ell_{\mathrm{wL1}}(\boldsymbol{H}, \boldsymbol{G}) = \frac{\sum_p (1 + \alpha_{\mathrm{csp}} G_p) \left| \sigma(H_p) - G_p \right|}{\sum_p (1 + \alpha_{\mathrm{csp}} G_p) + \epsilon}, \tag{S14}$$

where $\sigma(\cdot)$ is the sigmoid function, $\alpha_{\mathrm{csp}} = 5$ increases the contribution of locations near the target center, and $\epsilon = 10^{-6}$ is the numerical-stability constant used in the denominator. Here, p retains its earlier meaning as a spatial-location index. We define

$$\mathcal{L}_{\mathrm{cur}}^t = \ell_{\mathrm{wL1}}\left(\boldsymbol{H}_{\mathrm{cur}}^t, \boldsymbol{G}^t\right), \qquad \mathcal{L}_{\mathrm{pred}}^t = \ell_{\mathrm{wL1}}\left(\boldsymbol{H}_{\mathrm{pred}}^t, \boldsymbol{G}^{t+1}\right). \tag{S15}$$

For every search frame with an available subsequent sampled frame, the two terms are combined as

$$\mathcal{L}_{\mathrm{csp}}^t = 0.4\mathcal{L}_{\mathrm{cur}}^t + 0.6\mathcal{L}_{\mathrm{pred}}^t. \tag{S16}$$

Since no subsequent target is available for the final sampled search frame, only its current-state term is retained. Thus, the complete causal state-prediction objective is

$$\mathcal{L}_{\mathrm{csp}} = \frac{1}{N_f}\left[\sum_{t=1}^{N_f-1}\left(0.4\mathcal{L}_{\mathrm{cur}}^t + 0.6\mathcal{L}_{\mathrm{pred}}^t\right) + \mathcal{L}_{\mathrm{cur}}^{N_f}\right]. \tag{S17}$$

The larger weight assigned to $\mathcal{L}_{\mathrm{pred}}^t$ emphasizes short-term future-state prediction, while the current-state term stabilizes the spatial response of the temporal branch.

Target instance-consistency objective. The instance-consistency objective is applied to target representations obtained by Gaussian-weighted pooling. Specifically, the target region of the initial template is used to construct a normalized template representation $\boldsymbol{T}_i$, while the target region of the t-th search frame produces the normalized search representation $\boldsymbol{S}_i^t$. In the current implementation, the template representation is detached from the computation graph and serves as a fixed target anchor for the instance-consistency objective. For the i-th sample, $\boldsymbol{T}_i$ forms the positive template for $\boldsymbol{S}_i^t$, whereas the template representations of the other samples in the mini-batch serve as negative instances. The frame-level InfoNCE objective is

$$\mathcal{L}_{\mathrm{ins}}^t = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\left(\mathrm{sim}\left(\boldsymbol{S}_i^t, \boldsymbol{T}_i\right)/\tau\right)}{\sum_{j=1}^{B} \exp\left(\mathrm{sim}\left(\boldsymbol{S}_i^t, \boldsymbol{T}_j\right)/\tau\right)}, \tag{S18}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity, B is the batch size, and $\tau = 0.2$ is the temperature coefficient. Averaging over all sampled search frames gives

$$\mathcal{L}_{\mathrm{ins}} = \frac{1}{N_f}\sum_{t=1}^{N_f} \mathcal{L}_{\mathrm{ins}}^t. \tag{S19}$$

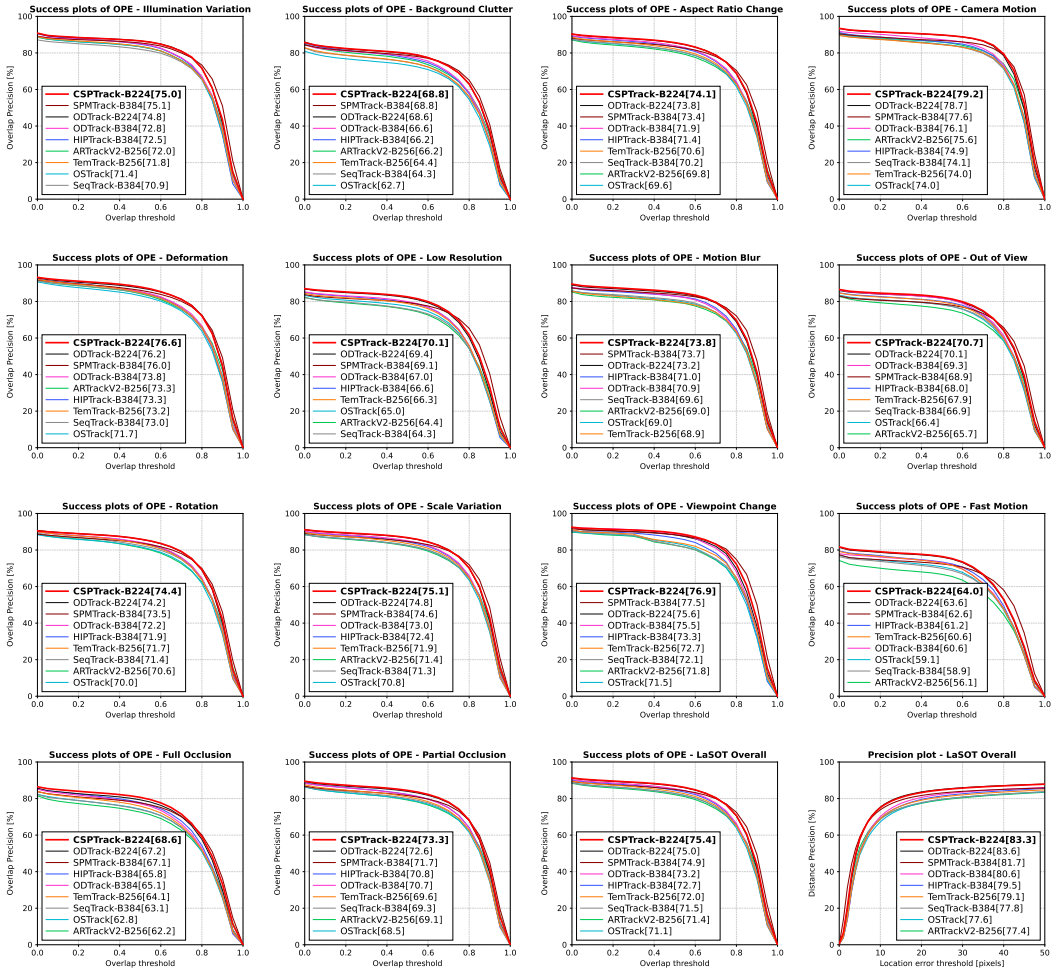

Figure S1: Attribute-wise AUC comparison on the LaSOT test set. Results are reported for all 14 challenge attributes, including fast motion, out-of-view, aspect-ratio change, partial occlusion, illumination variation, full occlusion, rotation, low resolution, background clutter, viewpoint change, scale variation, motion blur, deformation, and camera motion. "ALL" denotes the overall AUC on the complete LaSOT test set. CSPTrack-B224 achieves superior results across most attributes.

This objective pulls each search-target representation toward its corresponding template anchor while separating it from the target representations of other instances. Since it is used only during training, the instance-consistency objective introduces no additional inference cost. RaTP does not introduce an additional optimization objective. Its Ridge-reconstruction error is used to rank search tokens rather than being directly minimized during training. Consequently, token compression does not add another term to Eq. S9.

S3 Attribute-wise Analysis on LaSOT

To provide a more comprehensive evaluation beyond the overall tracking performance, we conduct an attribute-wise analysis on the LaSOT test set. LaSOT annotates each sequence with 14 challenge attributes, covering target motion, appearance variation, visibility degradation, and background interference. We compare CSPTrack-B224 with a broad range of representative purely visual trackers, as listed in the legend of Figure S1. The area under the success curve (AUC) is reported separately for each attribute.

As shown in Figure S1, CSPTrack-B224 achieves the highest AUC on 11 of the 14 LaSOT challenge attributes and ties for the best result on background clutter. For motion- and geometry-related chal-



CSPTrack-B224 SPMTrack-B384 HIPTrack-B384 ODTrack-B384 Ground Truth

(a) Qualitative comparison when targets undergo large deformation.



CSPTrack-B224 SPMTrack-B384 HIPTrack-B384 ODTrack-B384 Ground Truth

(b) Qualitative comparison when targets suffer from partial occlusion.



CSPTrack-B224 SPMTrack-B384 HIPTrack-B384 ODTrack-B384 Ground Truth

(c) Qualitative comparison when targets undergo large scale variation.

Figure S2: Visual comparison of CSPTrack with representative trackers on challenging LaSOT sequences. CSPTrack provides more accurate and stable target localization under substantial appearance changes and interference. The numbers in the upper-left corners indicate frame indices.

lenges, it achieves 79.2% AUC under camera motion, 75.1% AUC under scale variation, 74.4% AUC under rotation, 73.8% AUC under motion blur, and 64.0% AUC under fast motion. These results exceed the corresponding second-best AUC scores by 0.5, 0.3, 0.2, 0.1, and 0.4 percentage points, respectively. Its 76.9% AUC under viewpoint change is 0.6 percentage points below SPMTrack-B384. For appearance-related challenges, it obtains 76.6% AUC under deformation and 74.1% AUC under aspect-ratio change, exceeding the second-best AUC scores by 0.4 and 0.3 percentage points, respectively. Its 75.0% AUC under illumination variation is only 0.1 percentage points below the best result. Under visibility degradation, CSPTrack-B224 reaches 73.3% AUC under partial occlusion, 68.6% AUC under full occlusion, and 70.7% AUC under out-of-view, surpassing the second-best AUC scores by 0.7, 1.4, and 0.6 percentage points, respectively. It also achieves 70.1% AUC under low resolution, exceeding the runner-up by 0.7 percentage points, and matches the best background-clutter result at 68.8% AUC. Overall, CSPTrack-B224 achieves 75.4% AUC on LaSOT, exceeding ODTrack-B224 and SPMTrack-B384 by 0.4 and 0.5 AUC percentage points, respectively. These gains span motion, appearance, visibility, and background-related challenges.

S4 Qualitative Analysis under LaSOT Challenges

To complement the quantitative evaluation, we compare CSPTrack with representative trackers on challenging sequences from the LaSOT test split. Figure S2 presents three common tracking challenges:

- Figure S2(a) demonstrates the tracking results under large deformation, which requires adaptation to substantial shape and appearance changes.
- Figure S2(b) demonstrates the tracking results under partial occlusion, which requires preserving target identity from incomplete visual evidence.

- Figure S2(c) demonstrates the tracking results under large scale variation, which requires accurate localization under considerable target-size changes.

Each group contains two representative sequences, and the labels in the upper-left corners denote frame indices. Overall, the qualitative comparisons in Figure S2 show that CSPTrack maintains more stable target localization across the three representative challenge scenarios. Under large deformation, the target appearance changes substantially as its pose and shape evolve over time, weakening the correspondence between the current observation and the reference template. CSPTrack remains closely aligned with the ground-truth target throughout these variations, whereas several compared trackers exhibit noticeable localization drift or inaccurate target extents in challenging frames. Under partial occlusion, only a limited portion of the target remains visible in some frames, making target-background discrimination more difficult. CSPTrack preserves the target identity more consistently during occlusion and recovers accurate localization as the target becomes visible again, while competing trackers are more prone to position shifts, scale instability, or temporary drift. Under large scale variation, CSPTrack adapts the predicted bounding box more closely to the changing target extent, whereas some competing methods tend to underestimate or overestimate the target scale. These observations are consistent with the attribute-wise results in Figure S1, where CSPTrack shows strong performance under deformation, occlusion, and scale-related challenges. Qualitatively, the comparisons suggest that temporal state modeling contributes to stable target representations under rapid target changes, while the instance-consistency constraint helps preserve target discrimination when the available visual evidence becomes ambiguous. Together, these properties support reliable localization across substantial appearance changes, temporary visibility degradation, and pronounced target-scale variations.