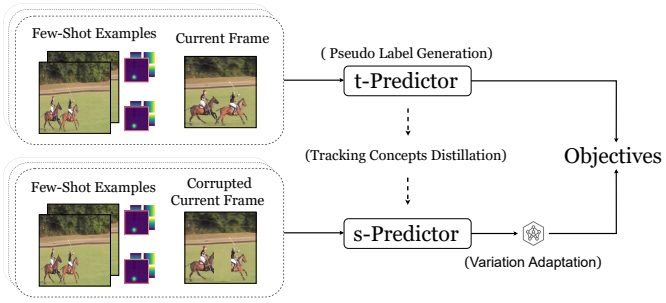
GOT-JEPA: Generic Object Tracking with Model Adaptation and Occlusion Handling using Joint-Embedding Predictive Architecture

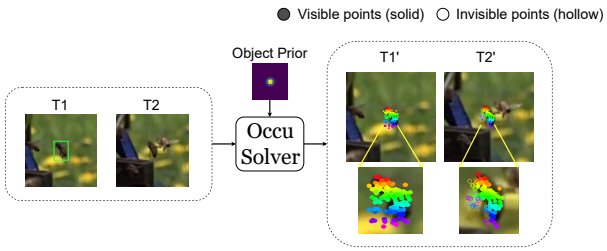
Shih-Fang Chen, Jun-Cheng Chen *Member, IEEE*, I-Hong Jhuo *Member, IEEE*, and Yen-Yu Lin *Senior Member, IEEE*

***Abstract*—The human visual system tracks objects by integrating current observations with previously observed information, adapting to target and scene changes, and reasoning about occlusion at fine granularity. In contrast, recent generic object trackers are often optimized for training targets, which limits robustness and generalization in unseen scenarios, and their occlusion reasoning remains coarse, lacking detailed modeling of occlusion patterns. To address these limitations in generalization and occlusion perception, we propose GOT-JEPA, a model-predictive pretraining framework that extends JEPA from predicting image features to predicting tracking models. Given identical historical information, a teacher predictor generates pseudo-tracking models from a clean current frame, and a student predictor learns to predict the same pseudo-tracking models from a corrupted version of the current frame. This design provides stable pseudo supervision and explicitly trains the predictor to produce reliable tracking models under occlusions, distractors, and other adverse observations, improving generalization to dynamic environments. Building on GOT-JEPA, we further propose OccuSolver to enhance occlusion perception for object tracking. OccuSolver adapts a point-centric point tracker for object-aware visibility estimation and detailed occlusion-pattern capture. Conditioned on object priors iteratively generated by the tracker, OccuSolver incrementally refines visibility states, strengthens occlusion handling, and produces higher-quality reference labels that progressively improve subsequent model predictions. Extensive evaluations on seven benchmarks show that our method effectively enhances tracker generalization and robustness. The code will be available at https://github.com/chenshihfang/GOT.**

***Index Terms*—Generic Object Tracking, Model Prediction, JEPA, Online Learning, Point Tracking, Few-Shot Learning**



(a) Learn to Build a Robust Model Predictor



(b) OccuSolver Allows Predicted Points to Become Object-Aware

Fig. 1. (a) GOT-JEPA extends the JEPA architecture to build a robust model predictor for model adaptation. The t-Predictor generates diverse tracking models for pseudo-labeling, while the s-Predictor predicts them using corrupted current frames. Both predictors use identical historical frames and tracking results as few-shot examples to aid target identification, while variations in the current frame drive robustness in tracking-model prediction. (b) OccuSolver enhances occlusion-handling capabilities. T1' shows initial point sampling. T2' shows that OccuSolver filters redundant points as invisible states, retaining essential points as visible states. This information is then utilized to enhance occlusion perception in the proposed GOT tracker.

I. Introduction

GENERIC Object Tracking (GOT) [1]–[3] allows machine visual tracking of any arbitrary target object specified by only an initial bounding box in the first frame and predicting its location in subsequent frames. Given this limited information, learning a robust tracker to accurately predict the target location in each frame of a dynamic environment is very challenging, especially in adverse conditions, where unseen target objects, complex distractors, and object deformations are present. Additionally, handling occlusions and out-of-view scenarios is crucial for long-term tracking.

Shih-Fang Chen and Yen-Yu Lin are with the Department of Computer Science, National Yang Ming Chiao Tung University (e-mail: csf.cs09@nycu.edu.tw; lin@cs.nycu.edu.tw) (Corresponding author: Y-Y Lin).

Jun-Cheng Chen is with the Research Center for Information Technology Innovation, Academia Sinica (e-mail: pullpull@citi.sinica.edu.tw).

I-Hong Jhuo is with Microsoft AI (e-mail: ihjhuo@gmail.com).

The tracking-by-detection paradigm [3] is widely used to learn trackers capable of handling dynamically changing targets in adverse scenarios [1], [4]. These trackers typically consist of a *model predictor* and a *localization head*. The model predictor dynamically updates the tracker so that the localization head of the updated tracker can better identify the target in the coming frames. To further address challenges such as occlusions and out-of-view scenarios, these trackers may estimate confidence scores and dynamically update their estimates. However, confidence scores are typically derived from appearance-based non-occlusion perception, whereas occlusion labels exhibit deficiencies in GOT, leading to unreliable confidence scores, especially when occlusions are present. The existing tracking-by-detection paradigm thus suffers from limited generalization and occlusion-handling capabilities.

Specifically, the model predictor of existing methods [4], [5] is prone to generating less reliable trackers when inferring targets unseen during training. This is because the existing

paradigm optimizes the model predictor during training to best identify learned tracking targets, rather than learning model prediction as a general skill. This limited learning paradigm hinders the model predictor's ability to adapt to unseen targets. To mitigate challenges such as occlusion and target disappearance, existing trackers have explored several directions. One line of research maintains multiple trajectories to track plausible targets and improve confidence reliability [6], [7]. Another line enhances occlusion robustness by promoting appearance invariance through masking strategies (e.g., masked autoencoders) [8], [9]. Alternative methods estimate target locations under low-confidence predictions to assist occlusion recovery [10] or incorporate visibility-aware confidence estimation [4], [5]. However, these methods share a common limitation in occlusion handling: they operate at the scene or box level and do not explicitly infer which target regions remain visible under partial occlusion. This limitation is further compounded by the scarcity of fine-grained occlusion annotations in generic object tracking, which leads to largely implicit supervision and limits the learning of detailed occlusion patterns.

Human cognition tracks objects by continuously integrating current observations with past information, adapting to changes in targets and environments, and reasoning about partial occlusion and visibility at fine granularity, even for unseen targets. In contrast, current object-tracking systems lack the abstract reasoning capabilities needed to handle complex occlusions and unseen targets in dynamic environments. To bridge this gap, we introduce GOT-JEPA, as shown in Fig. 1. Our work advances the JEPA (Joint-Embedding Predictive Architecture) paradigm [11] from image feature prediction to the novel task of tracking model prediction. In the model-predictive learning paradigm of GOT-JEPA, a teacher predictor generates pseudo-tracking models from a clean current frame, while a student predictor learns to predict these models from a corrupted current frame, with identical historical conditioning for both predictors. This objective encourages recovery of target–background discrimination under degraded observations, yielding a robust and discriminative model predictor that generalizes to unseen objects and diverse environments.

The proposed OccuSolver enhances occlusion perception in GOT by combining GOT-derived object priors with the point tracker [12], thereby integrating high-level object semantics with low-level point visibility. Since the point tracker is point-centric and relies on initial query points from the object bounding box in GOT, these points may come from both the target and the background, leading to uncertainty about which points are beneficial to GOT. We thus align the point tracker with GOT by refining its output using GOT-derived object priors and objectives. The object priors further guide dynamic point sampling, adapting to new viewpoints and surfaces not covered by the initial sampling. OccuSolver then estimates the point-wise visibility state of a tracked object and enables the JEPA-trained model predictor to generate more effective target-adaptive tracking models by accounting for point-wise visibility states. It turns out that a tight coupling between GOT and OccuSolver is created: The tracker provides OccuSolver with improved object priors to identify point visibility, while the identified visibility states, in turn, help generate better reference labels for tracking model adaptation.

The main contributions are summarized as follows: First, we introduce GOT-JEPA, a model-predictive learning framework that extends the JEPA paradigm from image-feature prediction to tracking-model prediction for streaming inputs. GOT-JEPA trains a model predictor in which a teacher predictor produces pseudo-tracking models from clean frames, and a student predictor learns to predict them from corrupted frames, with identical historical conditioning for both predictors. This design improves robustness and generalization to unseen targets and dynamic scene variations. Second, we propose OccuSolver, which equips generic object tracking with fine-grained occlusion reasoning by tightly integrating high-level semantic perception with low-level geometric cues. Specifically, OccuSolver makes a point tracker object-aware by incorporating GOT-derived object priors and transferring point-level visibility signals to GOT-JEPA. These target-centric visibility cues enhance occlusion reasoning and provide higher-quality reference labels for tracking-model adaptation, stabilizing subsequent model predictions, and improving recovery after reappearance. Extensive experiments on seven benchmarks show consistent gains under occlusion and deformation, with superior generalization to both in-distribution and out-of-distribution targets.

II. Related Work

In this section, we discuss several research topics relevant to the development of our method.

A. Tracking-by-Detection

Some trackers, such as [13]–[28], approach GOT as a tracking-by-detection task [3]. Recent trackers [1], [4] following this paradigm utilize paired reference images and labels alongside the current frame to generate a tracking model in the form of discriminative correlation filters. This tracking model is produced through a predictor derived using meta-learning techniques [29]–[31]. A classification and regression head is also included to localize the target in the current frame. Despite the meticulous architecture design, their model predictors still tend to produce unreliable tracking models because they are prone to guiding the model predictor to derive tracking models for familiar targets during training, limiting generalization capability to unseen scenarios. In contrast, we propose GOT-JEPA, which learns to predict trackers using diverse pseudo-tracking models and effectively generalizes to both in- and out-of-distribution targets.

B. Matching-based Tracking

Several trackers [2], [32]–[40] formulate GOT as similarity learning followed by matching [3]. They focus on models capable of discriminating and matching templates within the current frame. A deep network trained offline to learn a matching function between templates and search images is later deployed online for tracking. This paradigm has evolved from conventional CNN-based trackers [32], [41]–[54], to recent

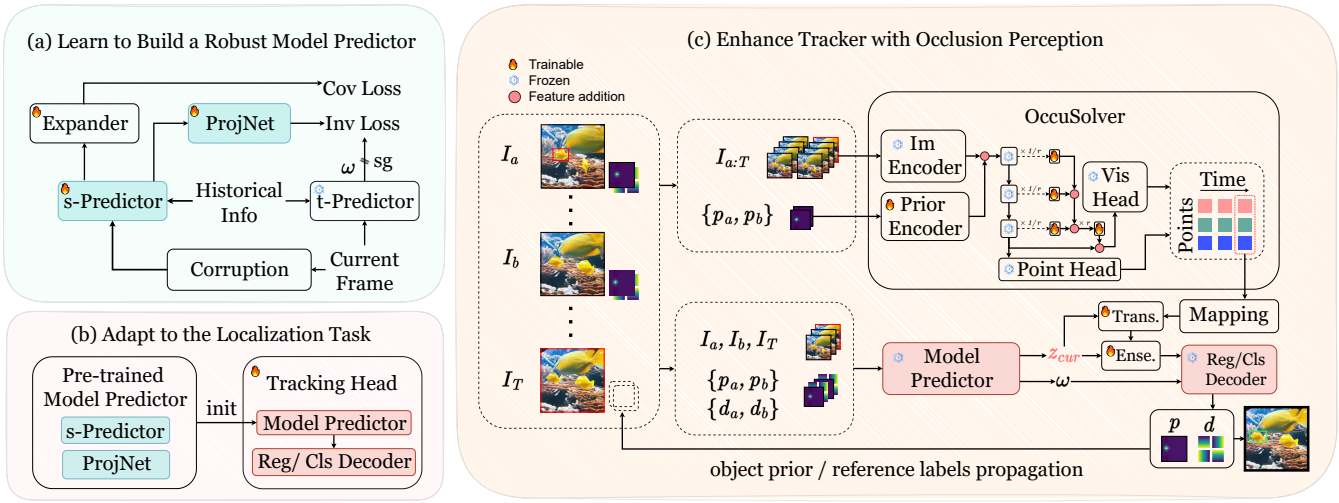


Fig. 2. **Overview of the proposed framework.** (a) We pre-train a robust model predictor using a JEPA-based approach. Conditioned on identical past information, a student predictor (s-Predictor) learns from a corrupted current frame to predict the tracking models generated by a teacher (t-Predictor) with the uncorrupted input. This process compels the student to learn representations that are robust to frame variations. Details are provided in Sec. III-B. (b) The pre-trained student predictor is then integrated into the tracking head with classification and regression decoders and fine-tuned for precise object localization. Details are provided in Sec. III-A and Sec. III-B. (c) To address occlusions, OccuSolver adapts a point tracker to be object-aware using priors from the object tracker. The resulting point visibility states are then integrated with visual features via an Ensemble Network, enabling the final model to better handle occluded targets and generate more accurate tracking models over time. Refer to Sec. III-C for details.

transformer-based trackers [51], [55]–[68]. Despite the improvements offered by Transformers in learning more discriminative matching functions or adopting advanced paradigms, such as auto-regressive [69]–[71], visual prompting [72], [73], masked image modeling [74]–[77], and diffusion [78], [79], these trackers exhibit limitations in generalization, compared to aforementioned model-prediction-based trackers, due to the inherent constraints of offline model training.

C. Occlusion Handling for Trackers

Researchers have developed various recovery and robustness strategies. Methods such as GlobalTrack [80], DeepMTA [6], and ARTrackV2 [7] perform global searches or maintain multiple candidate trajectories to re-detect targets after they disappear, while ensuring temporal trajectory consistency. SiamON [8] and ORTrack [9] employ masking techniques or synthetic occlusion generation to encourage the model to learn features that persist when parts of the object are hidden. Trackers such as ToMP [4] and PiVOT [5] use confidence score maps to detect object visibility and mitigate occlusion.

However, these approaches predominantly operate at a coarse granularity, treating the target as a single bounding-box entity. They typically rely on global confidence scores and cannot distinguish which specific parts of an object are visible or hidden during partial occlusion. In contrast, our method integrates the proposed OccluSolver to enable fine-grained, pixel-level occlusion perception. By explicitly estimating the visibility of discrete physical points using a point tracker [12] and applying our proposed point-tracker adaptation method to make these cues object-tracking-aware, our approach improves the object tracker's ability to distinguish occluded and unoccluded target sub-regions and maintain high precision even in complex scenarios.

D. Point Trackers

Point tracking introduced in TAP-Vid [81] aims to track arbitrary physical points on surfaces over a video. As TAP-Vid exhibits limitations in handling occlusions, in response, PIPs [82] and TAPIR [83] present point-tracking models that handle occlusions. These methods track points independently. OmniMotion [84] addresses this issue, but its test-time optimization is computationally inefficient. Cotracker [12] subsequently presents an efficient algorithm for tracking multiple points, while maintaining their correspondences. We modify Cotracker and integrate it into our OccuSolver to facilitate GOT. In contrast to the original Cotracker, our modified point tracker can leverage object priors provided by GOT, aligning its learning objectives with those of GOT. This enables the point tracker's knowledge to be adapted, helping GOT better handle partially occluded regions of the target object.

E. Joint-Embedding Predictive Architecture (JEPA)

JEPA [11] aims to predict the embedding of one signal from a compatible context signal. By performing prediction in representation space, JEPA can reduce reliance on irrelevant details and enable learning of transferable representations that improve transfer to downstream tasks. Building on this paradigm, I-JEPA [85] and V-JEPA [86] infer missing content in image representation space from corrupted inputs, emphasizing semantic cues over pixel-level details. S-JEPA [87] applies JEPA to 3D skeletons for action recognition, and BrainJEPA [88] extends JEPA to modeling of brain dynamics.

We adapt JEPA to tracking by shifting the traditional prediction target from semantic image representations to our proposed target-conditioned, discriminative tracking models. In our setting, the tracking model captures target–background discrimination conditioned on the observed history and the

current frame. We train the model predictor using the proposed GOT-JEPA learning paradigm: a teacher predictor generates pseudo-tracking models from an uncorrupted current frame, and a student predictor learns to predict them from a corrupted version, with identical historical conditioning for both predictors. This objective promotes consistent tracking-model prediction under diverse frame perturbations induced by data augmentation, encouraging a robust and discriminative model predictor that generalizes beyond target-specific optimization.

III. Method

In this work, we propose to enhance the generalization and robustness of a tracker through GOT-JEPA and OccuSolver. An overview of our method is illustrated in Fig. 2. Our tracker enhances the generalization capability by learning to predict tracking models within the proposed GOT-JEPA pretraining paradigm. OccuSolver further utilises object priors from GOT to adapt a point tracker, thereby enhancing its alignment with GOT for improved occlusion handling and enhanced tracking model prediction. We introduce the background of the tracking-by-detection paradigm in Sec. III-A. Then, we describe our GOT-JEPA in Sec. III-B and OccuSolver in Sec. III-C, respectively.

A. Background

In this work, we focus on tracking-by-detection-based trackers [1], [3]–[5] due to their enhanced adaptability. These trackers generally comprise an image encoder and a Tracking Head that consists of a model predictor, a regression decoder (RegDec), and a classification decoder (ClsDec).

The model predictor learns to predict the tracking models (or filters) which are used to detect and localize the target in the incoming frames. Specifically, the model predictor in [4], [5] processes the encoded features of the reference frames, the reference labels of the reference frames, and the current frame as inputs, where a frame is of feature resolution $H \times W$, label elements $p_a, p_b \in \mathbb{R}^{H\times W}$ are classification score map labels, and label elements $d_a, d_b \in \mathbb{R}^{H\times W}$ are regression score map labels.

To this end, the model predictor generates the refined current frame features $z_{cur} \in \mathbb{R}^{H\times W\times C}$ and the tracking model $\omega \in \mathbb{R}^{1\times C}$, where C is the number of channels. The ClsDec takes z_{cur} and ω as inputs, convolving them to output a predicted classification score map $p \in \mathbb{R}^{H\times W}$:

$$p = \omega * z_{cur}. \tag{1}$$

Then, a predicted regression map is computed using RegDec:

$$d = \text{RegDec}\left((\omega * z_{cur}) \cdot z_{cur}\right), \tag{2}$$

where operator $\cdot$ denotes channel-wise broadcasting multiplication, and RegDec uses four separate convolution layers to predict four feature maps $d \in \mathbb{R}^{H\times W\times 4}$ in the *ltrb* bounding box representation [89]. The coordinates with the highest score in p are mapped to the regression score map d for box coordinate prediction. These processes are used in [4], [5].

B. GOT-JEPA for Model Predictor Pre-training

In the tracking-by-detection paradigm, an accurate tracking model prediction is crucial for successful tracking. Our GOT-JEPA aims to maximize the model prediction capability at this stage by leveraging the JEPA framework [11] with objectives specialized for tracking. As shown in Fig. 2(a), our framework features a teacher network (t-Predictor) and a student network (s-Predictor). The s-Predictor incorporates a linear network (ProjNet) tail that learns to predict tracking models capable of accounting for frame variation, as mentioned in Sec. I.

To learn a robust model predictor, we initialize the teacher predictor (t-Predictor) from a pretrained tracking model predictor [4], [5] and keep it fully frozen during GOT-JEPA pretraining to generate pseudo-tracking models $\hat{\omega}$. We freeze the teacher because our objective uses teacher predictions from a clean current frame as pseudo supervision for a student trained on a corrupted current frame under the same history. A frozen teacher keeps pseudo targets stable and prevents the targets from drifting as the student changes, which reduces collapse risk from teacher–student co-adaptation. This principle is supported by recent frozen-teacher designs in self-distillation and representation learning/distillation [90]–[93].

The student predictor (s-Predictor) is trained to predict $\hat{\omega}$ from corrupted inputs. Both predictors share identical historical conditioning (*i.e.*, reference frames and reference labels), but differ in the current-frame observation: the teacher uses a clean frame, while the student uses its corrupted counterpart. This information asymmetry forces the student to match the same pseudo-tracking model under occlusions and distractors, thereby learning invariance and relying on stable target evidence. Since corruptions/augmentations are applied only to the student branch, the model is repeatedly trained on diverse corrupted views that correspond to the same teacher pseudo model, providing diverse supervision without updating the teacher. The lightweight projector in the student branch helps handle corruption-related variation and improves alignment to $\hat{\omega}$, strengthening robustness under adverse observations. The lightweight projector, termed ProjNet, is a linear network appended to the s-Predictor that generates the adaptive tracking model ω. Functioning like a hypernetwork [30], [94], [95], this predictor dynamically generates the weights used by the localization decoders (which we simplify as prediction tracking models in this paper). The student learns by aligning its generated model ω with the teacher's model $\hat{\omega}$ using the invariant loss $\mathcal{L}_{\text{inv}}(\cdot)$, which is computed over n batches:

$$\mathcal{L}_{\text{inv}}(\omega, \hat{\omega}) = \frac{1}{n}\sum_{i=1}^{n} \|\omega_i - \hat{\omega}_i\|_2^2. \tag{3}$$

Meanwhile, an additional tail, the Expander $Exp(\cdot)$, is appended to s-Predictor. It consists of a 1×1 convolution layer after s-Predictor and is used to expand the output channel while further optimizing the predictive capability of the tracking models with a covariance loss $\mathcal{L}_{\text{cov}}$ [96] as follows:

$$\mathcal{L}_{\text{cov}}(\omega_{\text{exp}}) = \frac{1}{c}\sum_{i\neq j} \left[\text{covM}(\omega_{\text{exp}})\right]_{i,j}^2, \tag{4}$$

where $\text{covM}(\omega_{\text{exp}}) = \frac{1}{n-1}\sum_{i=1}^{n}(\omega_{\text{exp},i} - \bar{\omega}_{\text{exp}})(\omega_{\text{exp},i} - \bar{\omega}_{\text{exp}})^T$ is the covariance matrix, ω_{exp} is the output of the Expander, i and j are the coordinates of the covariance matrix, $\bar{\omega}_{\text{exp}} = \frac{1}{n}\sum_{i=1}^{n}\omega_{\text{exp},i}$, $\omega_{\text{exp}} = \mathit{Exp}(\omega)$, and n represents the number of tracking models, each with a dimension of c for the input batches. This loss encourages the off-diagonal coefficients of $\text{covM}(\cdot)$ to approach zero, thereby reducing redundant information in the predicted tracking model. With large-scale, diverse tracking data, redundancy reduction encourages the model to learn more diverse and discriminative patterns, thereby improving the robustness and generalization of model predictions during GOT-JEPA pretraining. While prior works [96], [97] applies similar losses to representation learning, we utilize them for tracking-model prediction to enhance robustness to frame variation across arbitrary targets.

The overall objective function for GOT-JEPA, used in the tracking model prediction learning, is a weighted sum of the invariance and covariance terms:

$$\mathcal{L}_{\text{mp}} = \alpha\,\mathcal{L}_{\text{inv}}(\omega, \hat{\omega}) + \beta\,\mathcal{L}_{\text{cov}}(\omega_{\text{exp}}), \tag{5}$$

where α and β are hyperparameters controlling the importance of loss terms. Hyperparameter details are given in Sec. IV.

The model predictor enhances tracking model predictions by learning diverse prediction patterns from GOT-JEPA pretrained, extending beyond simple self-distillation [91], [93], [98]–[100]. Once the student model predictor is trained, as illustrated in Fig. 2(b), the classification and regression decoders (ClsDec and RegDec) are jointly fine-tuned for object localisation learning, where this stage validates whether the JEPA-trained model predictor benefits GOT.

C. OccuSolver

As depicted in Fig. 2(c), OccuSolver aims to enhance the GOT tracker by providing additional occlusion perception. This capability empowers our GOT-JEPA to further achieve superior tracking performance through pixel-level visibility information from the OccuSolver, consequently yielding higher-quality pseudo-reference labels for subsequent frames. It enables the model predictor of our GOT-JEPA to incrementally generate more accurate and refined tracking models.

OccuSolver uses object priors (*i.e.*, the reference labels) from the GOT tracker and an image sequence as its input. It then refines these priors with the Point Tracker [12] (*i.e.*, the frozen components depicted in OccuSolver of Fig. 2(c)), and subsequently generates a more accurate output from the Point Tracker that can benefit GOT.

Refining the point tracker to be object-centric. Given the point-centric nature and lack of object awareness in point trackers, where the initial query points are randomly sampled within the bounding box of the target object in the first frame of the sequence, adaptation is crucial for their utilization in GOT. To this end, we condition the Point Tracker with two object priors (p_a and p_b), identical to the reference labels used for the GOT trackers [4], [5]. The object priors are processed by a *Prior Encoder* (like the label encoder used in ToMP [4]), which encodes the object priors' features. These features are added element-wise to the image features of the first and middle frames in the Point Tracker sequence, respectively.

Specifically, for each query point in a frame sequence, OccuSolver first uses the image encoder of a pretrained point tracker (*i.e.*, Cotracker [12]) to compute the appearance features for each point on each frame. For query points across time, these features are concatenated spatially to form an appearance token $Q \in \mathbb{R}^F$. The appearance token is concatenated with the point track (*i.e.*, the points' coordinates), $PT \in \mathbb{R}^2$. The concatenated token serves as the input to an iterative transformer ($\mathit{iter\text{-}Trans}$), which acts as a non-trainable component positioned above the *Point Head*, and refines the query point coordinates PT and appearance features Q:

$$O(PT^{(m+1)}, Q^{(m+1)}) = \textit{iter-Trans}(PT^{(m)}, Q^{(m)}), \tag{6}$$

where $m = 1, 2, ..., M$ indexes the iteration, $PT^{(m)}$ and $Q^{(m)}$ are the refined point track and appearance features, respectively, and O represents the output tokens from the iterative Transformer. After iterations, the refined point track at the last iteration ΔPT is fed into a non-trainable *Point Head* for coordinate generation, while the refined appearance features ΔQ are passed to a learnable VisHead for visibility state estimation. Both the Point Head and VisHead, appended after the iterative Transformer, are adopted from CoTracker [12].

To enable the Point Tracker to utilize the object priors and benefit GOT, we pass each iterative points' appearance features through a four-head two-layer transformer ($\mathit{light\text{-}Trans}$). This network is fine-tuned using the Ladder Side Network [101], which is known for its efficient tuning of large networks and serves as the trainable component positioned above the *Point Head*. For each query point, the process is formulated as:

$$\hat{Q}^{(m)} = \mathit{light\text{-}Trans}(Q^{(m)}), \tag{7}$$

where $\hat{Q}^{(m)}$ is the output from the m-th iteration pass of $\mathit{light\text{-}Trans}$ (like the architecture used in [102], but with a reduced number of transformer heads and layers). For memory efficiency, outputs from each $\mathit{light\text{-}Trans}$ layer undergo dimension reduction before tuning. After this iterative process, the outputs are scaled by $\mathit{ScaleNet}$, a convolution network akin to MLP-Mixer's Mixer Layer [103], and conditioned on the iterative transformer's final output. The refined output Q_{cond} is then computed:

$$Q_{\text{cond}} = \hat{Q} + \Delta Q, \text{ where } \hat{Q} = \mathit{ScaleNet}\left(\sum_{m=1}^{M}\hat{Q}^{(m)}\right). \tag{8}$$

Finally, the refined Q_{cond} acts as the input to *VisHead* for visibility estimation. Together with the Point Head, OccuSolver predicts points' coordinates and their visibility statuses.

Connecting OccuSolver with GOT. While OccuSolver enhances the Point Tracker's ability to track objects, the underlying point tracking method faces two key limitations. First, it is prone to failure when objects move dramatically between frames, as each point is tracked within a constrained local search region. Second, for computational efficiency, we track a sparse set of query points (*e.g.*, 128 points), which can yield a coarse representation that is not fine-grained across the object's entire boundary. To mitigate these issues, we introduce an Ensemble Network, a four-head two-layer transformer architecture, that learns to modulate the sparse

point visibility features from OccuSolver with the dense visual features of the current frame from the GOT, creating a more robust and complete target representation.

We first introduce a *Mapping Function* to transform discrete OccuSolver point coordinates into a dense spatial representation. For each of the C predicted point coordinates, a Gaussian kernel is applied to generate an initial energy map e. Details of the Gaussian function generation can be found in [25]. If a point is invisible, its map is negated $(1-e)$. These individual energy maps are concatenated to form $\mathbf{E} \in \mathbb{R}^{H\times W\times C}$, which is then spatially concatenated with the GOT current frame feature z_{cur} and processed by a lightweight transformer (*i.e.*, the aforementioned Ensemble Network). This step models the interplay between visual and visibility cues, yielding a visibility-aware current frame features $\tilde{\mathbf{E}} \in \mathbb{R}^{H\times W\times C}$.

The *Ensemble Network*, $\mathcal{E}(\cdot,\cdot)$, then modulates between the original current frame feature $\mathbf{z}_{\text{cur}}$ and the visibility-aware feature $\tilde{\mathbf{E}}$. Here, $\mathbf{z}_{\text{cur}}$ denotes the current frame feature generated by the *Model Predictor*, as shown in the figure. This produces a refined feature $\tilde{z}_{\text{cur}} = \mathcal{E}(\tilde{\mathbf{E}}, z_{\text{cur}})$. The training objective for OccuSolver is defined as follows:

$$\begin{aligned}\mathcal{L} = \lambda_{\text{cpt}}\mathcal{L}_{\text{cls}}(\hat{p}, p_{\text{pt}}) + \lambda_{\text{rpt}}\mathcal{L}_{\text{reg}}(\hat{d}, d_{\text{pt}}) \\ + \lambda_{\text{cgot}}\mathcal{L}_{\text{cls}}(\hat{p}, p_{\text{got}}) + \lambda_{\text{rgot}}\mathcal{L}_{\text{reg}}(\hat{d}, d_{\text{got}}),\end{aligned} \tag{9}$$

where the total loss is weighted by λ_{cpt}, λ_{rpt}, λ_{cgot}, and λ_{rgot}.

Similar to Eq. (1) where the score map p is obtained by $p = \omega * z_{\text{cur}}$, z_{cur} is replaced by $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{z}}_{\text{cur}}$ in Eq. (9) to obtain p_{pt} and p_{got}, respectively. Similarly, in Eq. (2), the regression map d is given by $d = \text{RegDec}\left((\omega * z_{\text{cur}}) \cdot z_{\text{cur}}\right)$. Here, z_{cur} is substituted by $\tilde{\mathbf{E}}$ and $\tilde{z}_{\text{cur}}$ to generate d_{pt} and d_{got}. The tracking model ω for the above process is generated by the model predictor. This dual-supervision strategy ensures that the visibility features learned from OccuSolver are effectively integrated into the generic object tracker, enhancing the tracker's robustness to occlusion and deformation.

IV. Experiments

This section evaluates our method, covering experimental settings, SOTA comparison, and ablation studies.

A. Experimental Setting

Training Data. Like most trackers, e.g. [4], [5], [69], [114], we adopt the training splits of LaSOT, GOT10k, TrackingNet, and COCO for model training. The training data rigorously follows the VOT2022 challenge and GOT-10K guidelines.

Test Data. We use the following datasets for evaluation:

- **AVisT** [115]: It is designed for testing without a training set, encompassing 120 short and long sequences, averaging 664 frames each under adverse visibility conditions.
- **NfS** [116] and **OTB-100** [117]: They are used for testing without a corresponding training set, each containing 100 sequences with an average of 534 frames per sequence.
- **GOT-10k** [104]: It has 420 short sequences with an average of 149 frames per sequence, featuring non-overlapping object classes in the training and test sets.
- **LaSOT** [118] and **TrackingNet** [118]: They provide training data where test classes fully overlap with training classes. LaSOT provides 280 long sequences with an average of 2k frames per sequence TrackingNet offers 511 short sequences, averaging 471 frames each.
- **VOT2022** [119]: It is the 2022 edition of the Visual Object Tracking short-term box (VOT-STb2022) challenge.

Evaluation Metrics. Like recent methods [5], [114], we evaluate trackers using the following metrics:

- **SUC** (success rate): The percentage of frames in which the predicted bounding box overlaps the ground truth by at least an IoU threshold or the average of all thresholds.
- **Pr** (precision): It measures the percentage of frames where the predicted target center is within T pixels of the ground-truth center. T is set to 20 in this work.
- **NPr** (normalized precision): It is the percentage of frames where the center location error normalized by the target's box diagonal is less than a threshold 0.2.
- **AO** (average overlap): It represents the mean IoU between the predicted and ground-truth bounding boxes.

Training Procedure. Our method involves two main stages: 1) GOT-JEPA training and 2) OccuSolver training.

At Stage 1, we first pre-train the Model Predictor using our JEPA architecture. After pre-training, it is fine-tuned with the Tracking Head. Specifically, during pre-training, a student predictor learns to predict tracking models from corrupted inputs. A teacher predictor, derived from the ToMP-L (a ViT-L variant of ToMP [4], [5]) model predictor, provides pseudo-tracking models from non-corrupted inputs. This setup allows the student predictor to learn from superior pseudo-tracking models and adapt to more challenging input cases.

The pre-training stage focuses solely on pre-training the model predictor to generate tracking models, without involving the Tracking Head or localisation objectives. After the Model Predictor is pre-trained, the Tracking Head is trained jointly. The tracking performance of GOT-JEPA without OccuSolver is then assessed (Table V, row (4)).

At stage 2, we train OccuSolver with the model predictor and Tracking Head. The model predictor and Tracking Head are initialized with weights trained at stage 1, and their model weights are frozen in this stage. This stage investigates whether OccuSolver improves the tracker compared to the tracker trained at stage 1. OccuSolver processes a 8 consecutive frames from a sequence. The Tracking Head treats frames 1 and 5 as the reference frames, and frame 8 as the current frame, along with their corresponding labels.

Implementation Details. Our method is implemented using PyTorch 2.0.0 and CUDA 11.7, and the tracker is built upon the ToMP framework [4]. The tracker operates on an NVIDIA RTX 4090 GPU, utilizing approximately 3 GB of GPU memory during evaluation and achieving 24 FPS and 50 FPS for the high- and low-resolution variants, respectively. In the first training stage, 8 GPUs are employed to train the proposed tracker with the L-378 variant. In the second stage, 4 GPUs are used. For the L-252 variant, all training stages are conducted using 4 GPUs. The batch size ranges from 48 to 64, maximized according to available computational resources.

TABLE I
COMPARISON OF OUR METHOD WITH SOTA TRACKERS ON VARIOUS DATASETS. ALL TRACKERS UTILISE THE SAME TRAINING DATA FOR CONSISTENCY. "*" INDICATES ADHERENCE TO SPECIFIC GOT-10K (GOT) GUIDELINES [104]. FOR EXISTING METHODS, WE REPORT THEIR HIGHEST-RESOLUTION VARIANT, PRIORITISING VIT-L BACKBONES WHEN AVAILABLE, OTHERWISE THEIR BEST-PERFORMING VARIANT.

Training-Test Class Overlap		Low or No Overlap				Full Overlap				
Dataset		AVisT	NfS	OTB	GOT*	LaSOT			TrackingNet	
Tracker	Venue	SUC	SUC	SUC	AO	NPr	Pr	SUC	NPr	SUC
GOT-JEPA (Ours)	-	**63.7**	**70.8**	**73.2**	**79.6**	**85.3**	**83.2**	**75.4**	**90.6**	**86.4**
UniSOT [105]	TPAMI26	57.8	67.6	-	-	-	78.3	71.3	-	84.1
SAMURAI [106]	TIP26	-	69.2	71.5	81.7	82.7	80.2	74.2	-	85.3
PiVOT [5]	TMM25	62.2	68.2	71.2	76.9	84.7	82.1	73.4	90.0	85.3
CVT-Track [107]	TCSVT25	-	-	-	74.9	80.4	77.9	71.2	88.2	84.2
MPIT [108]	TCSVT25	-	66.9	70.9	73.3	78.2	73.7	69.4	88.1	83.3
USCLTrack [109]	TCSVT25	-	-	70.8	75.8	82.4	89.9	73.5	89.4	84.9
MFDSTrack [110]	TCSVT25	-	66.8	66.8	73.6	81.2	77.6	72.1	88.5	84.2
MGTrack [111]	TCSVT25	-	-	-	76.2	81.0	84.6	84.7	89.7	84.7
SATrack [112]	IJCV25	58.4	67.5	-	75.4	81.4	78.4	72.0	89.0	84.7
SuperSBT [113]	TPAMI24	-	67.7	68.9	75.5	82.5	78.6	72.8	88.9	84.8
LoRAT [114]	CVPR24	62.0	66.7	72.0	77.5	84.1	82.0	75.1	89.7	85.6
ARTrackv2 [7]	CVPR24	-	68.4	-	79.5	82.8	81.1	73.6	90.4	86.1
HIPTrack [73]	CVPR24	-	68.2	71.0	74.5	82.9	79.5	72.7	89.1	84.5
SiamON [8]	TCSVT23	-	-	64.4	-	-	-	-	-	-
ROMTrack [59]	ICCV23	59.1	67.5	71.4	74.2	81.4	78.2	71.4	89.0	84.1
GRM [58]	CVPR23	54.5	66.9	68.9	73.4	81.2	77.9	71.4	88.9	84.0
OSTrack [57]	ECCV22	57.7	66.5	68.1	73.7	81.1	77.6	71.1	88.5	83.9
ToMP [4]	CVPR22	50.9	67.0	70.1	-	79.2	73.5	68.5	86.4	81.5

TABLE II
COMPARISONS AMONG DIFFERENT TRACKERS ON VOT-STB2022. SOTA RESULTS ARE REPORTED FROM THE VOT2022 PAPER [119].

Metrics	GOT-JEPA	MixFormerL	OSTrackSTB	TransT_M	SwinTrack	tomp
Robustness	0.898	0.859	0.869	0.849	0.803	0.818
AUC	0.728	0.708	0.680	0.639	0.626	0.628

TABLE III
ATTRIBUTE-WISE RESULTS ON OTB, AVISTT, AND LASOT USING LARGE-RESOLUTION VARIANTS FOR SEVERAL IMPORTANT ATTRIBUTES.

Dataset	Tracker	Background Clutter	Deformation	Occlusion	Out-of-View	Target Visibility
OTB	**Ours**	71.4	69.5	70.3	72.5	–
	ToMP-L	68.2	67.1	67.3	69.4	–
	PiVOT	68.4	68.6	67.8	66.1	–
	ROMTrack	67.4	68.0	68.9	69.5	–
AVisT	**Ours**	67.2	49.0	61.3	–	67.8
	ToMP-L	65.5	45.7	57.4	–	65.2
	PiVOT	67.1	47.7	59.2	–	66.5
	MixFormer	59.6	43.5	54.8	–	61.2
LaSOT	**Ours**	69.6	76.6	70.7	70.3	–
	ToMP-L	66.4	74.9	67.1	68.3	–
	PiVOT	67.3	74.9	68.3	69.3	–
	ROMTrack	64.6	73.4	66.9	67.1	–

In Eq. (5), α is 25 times higher than β. In Eq. (9), we set $\lambda_{\text{cgot}} = 200$, $\lambda_{\text{cpt}} = 100$, $\lambda_{\text{rgot}} = 1$, and $\lambda_{\text{rpt}} = 0.5$, respectively. We use Copy-Paste [120], [121] to simulate frame variants such as occlusions and distractor elements. The copy-paste operation is applied in feature space on the patch-feature grid of the current frame. Concretely, after the backbone produces a feature map F with shape $B \times C \times H \times W$, we sample a corruption ratio $\rho \sim \mathcal{U}(0, \rho_{\max})$ with $\rho_{\max} = 0.2$ and set $K = \lfloor \rho HW \rfloor$. For each training sample, we randomly select K patch positions on the $H \times W$ grid as the source positions to copy, and randomly choose another K patch positions on the same grid as the target positions to paste. This ensures that the copied-and-pasted regions vary across samples and iterations. The features at the target positions are replaced with those copied from the source positions. This corruption is applied only to the student branch for the current-frame features, while the teacher branch uses clean current-frame features. AdamW [122] is used as the optimization solver. The learning rate is set to 10^{-4} for most components, while the Expander of the s-Predictor uses a rate of 10^{-3}.

The objective function for target classification is the compound hinge loss of DiMP [1], while the GIoU loss [123] is used for target regression. In line with recent works such as PiVOT [5] and LoRAT [114], we employ ViT-L as the backbone for image feature extraction, utilizing weights pretrained with DINOv2 [92]. The backbone remains frozen during tracker training. These settings are used in all experiments. Further details are in the supplementary material. To mitigate the computational cost of using higher image resolutions, as is common practice in recent works [69], [114], [124], [125], we use lower resolutions for ablation studies but employ higher resolutions for comparison with SOTAs: 1) **GOT-JEPA-252**, where the frame resolution and the patch token size are respectively set to 252×252 and 18×18, and 2) **GOT-JEPA-378**, where the frame resolution and the token size are respectively set to 378×378 and 27×27.

B. Comparisons with the State-of-the-Art Methods

Tab. I compares GOT-JEPA with SOTAs on several benchmark datasets. 'ToMP-L' is the baseline method that utilises a high image resolution. Our tracker is mostly close to PiVOT [5], where both our tracker and PiVOT use the DiNOv2 [92] ViT-L variant of ToMP [4] (*i.e.*, ToMP-L) as a base tracker. LoRAT [114] also uses the DiNOv2 ViT-L for

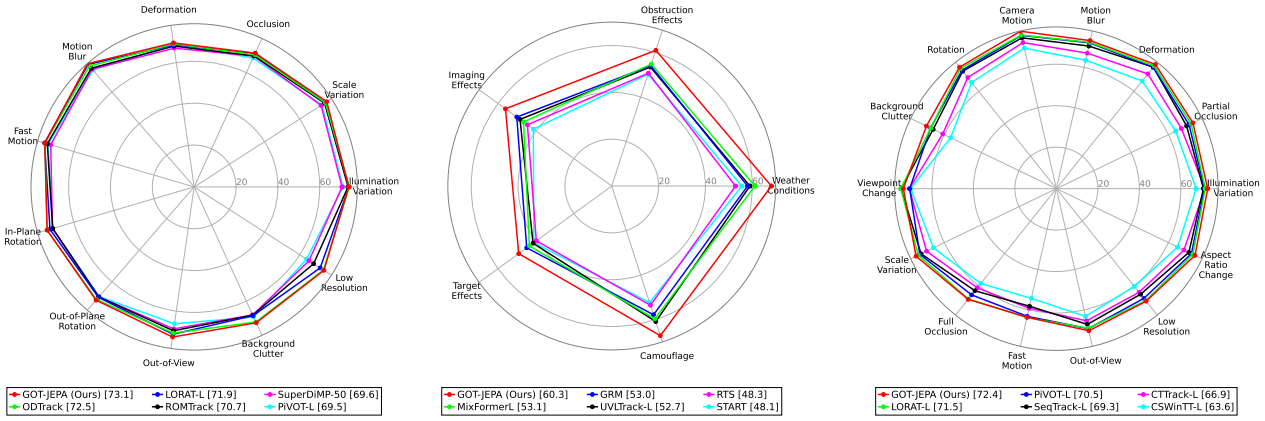


Fig. 3. This figure presents the attribute analysis of OTB-100, AVisT, and LaSOT from left to right, with the average scores at the bottom.

TABLE IV
COMPARISONS AMONG DIFFERENT TRACKERS ON MULTIPLE DATASETS USING OP50 AS A METRIC.

Dataset	NfS	AVisT	LaSOT
Tracker / Metric		OP50	
Ours	89.60	73.67	86.46
ToMP-L	85.69	72.55	84.78
LoRAT-L	85.56	-	85.11
UVLTrack-L	85.58	65.11	83.14
GRM-L	83.53	63.83	82.75
SeqTrack-L	82.37	-	82.98

image feature, though it uses OSTrack [57] as a base tracker. We also compare our tracker against several other ViT-L-based trackers, including DiffusionTrack [79], SeqTrack [69], GRM [58], UVLTrack [126], CTTrack [74], CSWinTT [127], and ODTrack [128]. Overall, GOT-JEPA exhibits robust performance and generalizes well to out-of-distribution and in-distribution targets.

As shown in Tab. I, GOT-JEPA demonstrates strong generalization against recent state-of-the-art trackers on both out-of-distribution and in-distribution benchmarks. For out-of-distribution evaluation, GOT-JEPA achieves a success rate of 63.7% on AVisT, outperforming PiVOT (62.2%), LoRAT (62.0%), and UniSOT (57.8%). On NfS, GOT-JEPA reaches 70.8%, exceeding HIPTrack and PiVOT (68.2% each) and UniSOT (67.6%). On OTB, GOT-JEPA attains the best success rate of 73.2%, surpassing SAMURAI (71.5%), LoRAT (72.0%), ROMTrack (71.4%), and PiVOT (71.2%).

On GOT-10k, GOT-JEPA achieves the highest AO of 79.6%, outperforming LoRAT (77.5%), PiVOT (76.9%), and MG-Track (76.2%). However, our tracker lags behind SAMURAI, which uses SAM 2 for segmentation tracking, while our tracker still performs better than the other trackers across the remaining datasets, except on GOT-10K. These consistent gains under distribution shift validate the effectiveness of OccuSolver, which leverages object-aware visibility cues for explicit occlusion handling. Furthermore, the results highlight the benefit of GOT-JEPA pre-training in enhancing target-identity preservation over long temporal windows, particularly under adverse visibility conditions.

For in-distribution evaluation, GOT-JEPA remains competitive and achieves the best normalized precision on LaSOT (85.3% NPr), while also attaining strong accuracy in success rate (75.4%). On TrackingNet, GOT-JEPA achieves the best results in both metrics, with 90.6% NPr and 86.4% success rate, outperforming strong baselines such as LoRAT (89.7% NPr and 85.6% success rate) and PiVOT (90.0% NPr and 85.3% success rate).

The comparison on the VOT2022 challenge, as shown in Tab. II, further demonstrates that the tracker achieves the highest robustness and AUC scores.

Comparison of Trackers Using OP50 as a Metric: In addition to SUC, NPr, and Pr, we compare trackers using OP50, which measures the percentage of frames where the predicted and ground-truth IoU exceed 50%. The results are presented in Tab. IV. On the NfS dataset, our tracker surpasses UVLTrack-L by 4.02%. On the AVisT dataset, it outperforms UVLTrack-L by 8.56%. On the LaSOT dataset, it exceeds LoRAT-L by 1.35%.

NPr, Pr, and SUC Plots and Analysis: We report the Normalized Precision (NPr), Precision (Pr), and Success (SUC) plots on four datasets: NfS, AVisT, LaSOT, and OTB-100. Other datasets, including VOT2022, TrackingNet, and GOT-10k, are evaluated on online servers without plot outputs and are therefore excluded from this analysis. The generation of plots requires raw tracker outputs; hence, methods lacking such data are omitted from the comparison.

In the Precision (Pr) and Normalized Precision (NPr) plots, the x-axis represents thresholds, while the y-axis denotes the percentage of frames where the predicted target center lies within these thresholds. Trackers are typically ranked at 20 pixels in Pr or 0.2 in NPr. In the Success (SUC) plot, the x-axis corresponds to IoU thresholds, and the y-axis shows the percentage of frames that exceed these thresholds. Trackers are ranked by the average precision across all thresholds.

The detailed analysis of each dataset is summarized as follows:

NfS: As illustrated in Fig. 4, our tracker outperforms all competing methods once the threshold exceeds 0.1 in NPr or 10 pixels in Pr. For SUC, the proposed method consistently achieves superior performance across all thresholds.

AVisT: As shown in Fig. 5, the AVisT dataset, which is

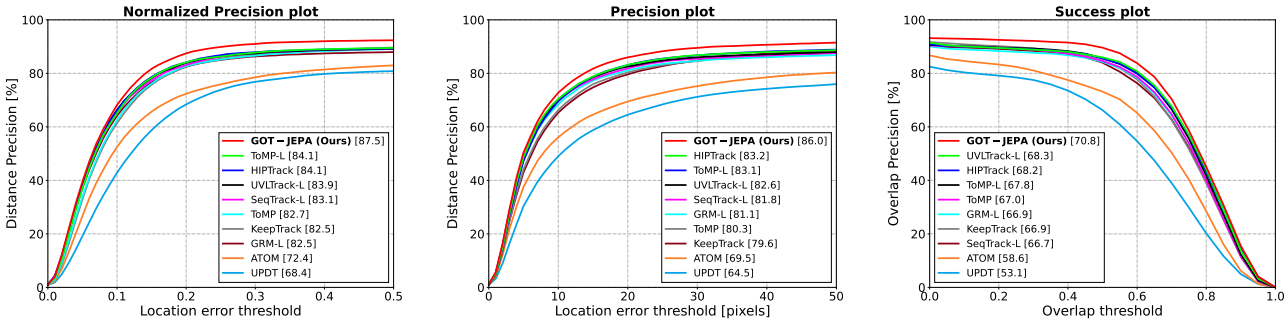


Fig. 4. **Comparison of methods using NPr, Pr, and SUC plots on the NfS dataset, from left to right.**

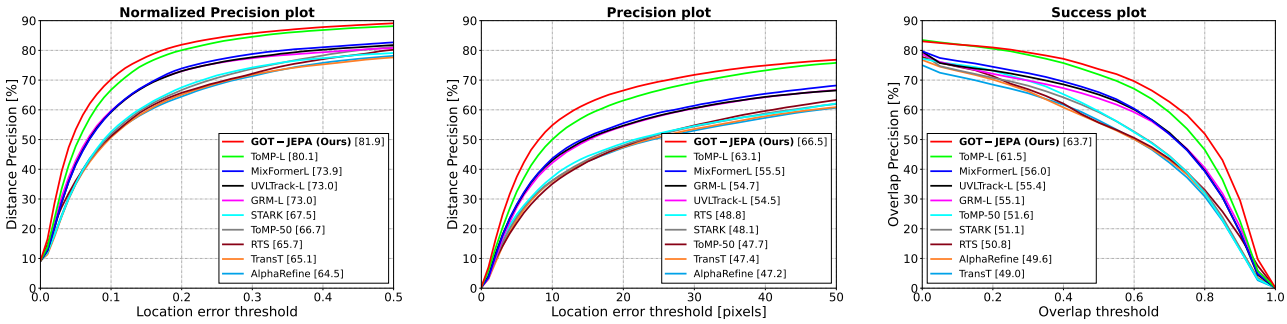


Fig. 5. **Comparison of methods using NPr, Pr, and SUC plots on the AVisT dataset, from left to right.**

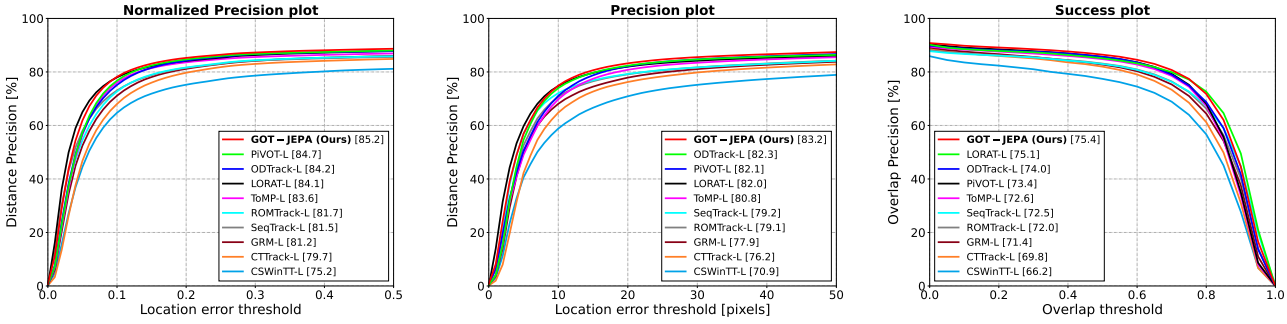


Fig. 6. **Comparison of methods using NPr, Pr, and SUC plots on the LaSOT dataset, from left to right.**

training-free and includes diverse adverse scenarios, demonstrates that our tracker consistently surpasses all baselines across metrics and thresholds, even under challenging conditions (NPr $<$ 0.1 or Pr $<$ 10 pixels).

LaSOT: In Fig. 6, on this in-distribution dataset, our tracker slightly surpasses the state-of-the-art method in Pr and shows a clear advantage in NPr, particularly around the 0.1 threshold, indicating robustness under size-normalized evaluation. For SUC, it outperforms most competing trackers once the threshold exceeds 0.6.

C. Attribute Analysis

Attribute-Based Analysis: We conduct an attribute-based analysis by comparing our method with several state-of-the-art trackers [4], [5], [48], [55], [56], [58], [59], [69], [72]–[74], [126], [126]–[132] using radar plots, as shown in Fig. 3. This analysis reveals the strengths and weaknesses of different methods and highlights potential areas for improvement. It is worth noting that attribute-based analysis requires the raw tracking results of each method. If the raw data of a tracker are unavailable, or if a dataset lacks an attribute analysis protocol (*e.g.*, , third-party servers without attribute results), those trackers are excluded from the analysis.

We also provide detailed numeric attribute results for key attributes across three datasets using the large-resolution variants in Table III. For ToMP-L [4], [5], this is the baseline tracker with a DINOv2 ViT-L backbone. PiVOT [5] uses the same baseline tracker as our method but additionally incorporates CLIP [133] as a semantic feature. ROMTrack [59] and MixFormer [132] use different baseline trackers, so we report their results for reference. Since ROMTrack does not report results on AVisT, we include MixFormer results as another variant of a similar method for comparison.

OTB-100: As shown in Fig. 3 (first column), most trackers achieve comparable performance, whereas our tracker exhibits superior overall results.

AVisT: As illustrated in Fig. 3 (second column), our tracker achieves significant improvements across all attributes compared with other trackers, demonstrating its effectiveness in

TABLE V
ABLATION STUDIES ON TRACKER COMPONENTS WERE EVALUATED USING SUC ON THREE DATASETS UNDER GOT-JEPA-252.

	JEPA-Pretrain		OccuSolver	AVisT	LaSOT	OTB
	Inv. Loss	Cov. Loss				
(1)	-	-	-	59.2	70.7	69.2
(2)	-	-	✓	59.6	71.1	69.5
(3)	✓	-	-	60.4	72.3	70.2
(4)	✓	✓	-	60.8	72.6	70.3
(5)	✓	✓	✓	61.7	73.4	70.6

handling unseen and challenging scenarios.

LaSOT: As depicted in Fig. 3 (third column), aided by GOT-JEPA and OccuSolver, our tracker exhibits superior robustness against object deformation, scale variation, and partial invisibility, while maintaining resilience to background clutter when processing in-distribution sequences such as those in LaSOT.

TABLE VI
EFFECT OF INVARIANCE AND COVARIANCE LOSS WEIGHTS IN GOT-JEPA PRE-TRAINING (L-252, WITHOUT OCCUSOLVER).

Regularization Method	α	β	AVisT	LaSOT
Inv	1	0	60.4	72.3
Inv + Cov	1	1	60.1	72.1
Inv + Cov	10	1	60.6	72.5
Inv + Cov	25	1	60.8	72.6
Inv + Cov	50	1	60.7	72.6

D. Ablation Studies

We conduct several ablation studies in this subsection to validate the effectiveness of each proposed component. The baselines used in our ablation study include ToMP-L [5], which is the DINOv2 [92]-based ViT-L variant of ToMP [4] that utilises the same backbone. Our tracker is built upon the same framework as ToMP-L.

Effect of Pretraining, Covariance Loss, and OccuSolver: Tab. V presents the ablation studies on GOT-JEPA, the covariance loss, and OccuSolver across five configurations and three datasets. The performance difference between the baseline in row (1) and the full model in row (5) indicates that GOT-JEPA enhances the ability of the tracker to generate more discriminative features and perform better occlusion reasoning, thereby improving overall tracking performance. When evaluating OccuSolver alone, a moderate performance gain is observed compared with the baseline tracker (row (1) vs. row (2)). However, when OccuSolver receives input from the JEPA-pretrained tracker (row (4) and row (5)), the improvement becomes substantially larger. This is because the JEPA-pretrained tracker provides higher-quality pseudo labels as object priors, enabling OccuSolver to infer more accurate visibility information, which supports generic object tracking. Moreover, the configuration "JEPA-Pretrain" with only the invariance loss (row (3)) already yields a noticeable improvement over the baseline, and the incorporation of the covariance loss (row (4)) further enhances the performance by 1.6%. Overall, compared with the baseline tracker, the proposed pretraining strategy combined with OccuSolver leads to performance gains of 2.5% on AVisT, 2.7% on LaSOT, and 1.4% on OTB-100.

TABLE VII
ABLATION STUDIES ON THE IMPACT OF PROPOSED COMPONENTS WITH VARYING ATTRIBUTES ON THE AVISTT [115] DATASET. THE BOTTOM ROW HIGHLIGHTS MORE SPECIFIC DESCRIPTIONS OF THE ATTRIBUTES.

Tracker	JEPA Pretrain	Occu Solver	Obstruction Effects	Target Effects	Imaging Effects	Weather Conditions	Camouflage
ToMP-L	-	-	56.74	42.84	54.06	62.39	65.00
GOT-JEPA	✓	-	58.75	46.66	54.94	66.08	62.42
GOT-JEPA	✓	✓	61.86	50.59	57.44	65.83	62.45
Attribute Descriptions			Occlusion	Distractor Deformation	Low-light Archival	Target-Visibility	Background-Clutter

TABLE VIII
EVALUATION OF CORRUPTION TYPES AND THEIR IMPACT ON PRE-TRAINING STRATEGIES UNDER GOT-JEPA-252.

	JEPA-Pretrain	Copy-Paste	Masking	AVisT	LaSOT
(1)				59.2	70.7
(2)		✓		59.5	71.2
(3)	✓	✓		60.8	72.6
(4)	✓		✓	60.0	71.3

TABLE IX
EVALUATING THE IMPACT OF PROJNET ON THE TRACKER. GOT-JEPA IS EVALUATED AT A RESOLUTION OF 252 WITHOUT OCCUSOLVER.

Tracker		ProjNet	AVisT	LaSOT	OTB-100
(1)	ToMP-L	-	59.2	70.7	69.2
(2)	ToMP-L	✓	58.9	70.5	69.2
(3)	GOT-JEPA	-	59.6	71.4	69.3
(4)	GOT-JEPA	✓	60.8	72.6	70.3

Table VI reports results for GOT-JEPA-252 without OccuSolver, isolating the effects of the invariance and covariance objectives. The invariance term is necessary for learning from tracking-model predictions, while the covariance term further improves the predictive capability of the tracking models and encourages diverse prediction patterns across frame variants. The ratio 25:1 provides the best overall trade-off. When the covariance term is overemphasized, as in the 1:1 ratio, performance drops on both AVisT and LaSOT, suggesting that an overly strong covariance objective can hinder invariance-driven prediction learning. When the covariance term is set to a moderate level, such as 10:1 or 50:1, performance is consistently higher than the invariance-only baseline 1:0, indicating that the covariance term is most effective as a regularizer rather than a dominant loss. Notably, AVisT does not provide training data, so the improvement under ratio-based settings with the covariance loss suggests stronger out-of-distribution generalization. Meanwhile, although LaSOT provides training data, the covariance loss still yields consistent gains, supporting its benefit for the proposed model-adaptation pretraining.

Furthermore, Tab. VII analyzes how the proposed components affect individual tracking attributes. The model predictor, pre-trained with JEPA, enhances the tracker's ability to handle distractors, deformation, and reduced target visibility in adverse weather conditions, such as rain or fog. OccuSolver further enhances performance in scenarios with occlusion and deformation.

Corruption Types for GOT-JEPA Pretraining. We employ copy-paste augmentation as the primary corruption technique and also evaluate masking, as reported in Tab. VIII. Row (1) shows the ToMP-L baseline at a resolution of 252, while row (2) introduces copy-paste without JEPA pre-training.

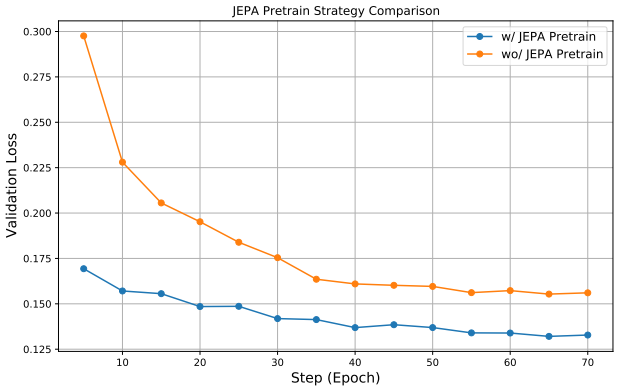


Fig. 7. An analysis of the validation curve to study how tracker pre-training, with or without JEPA, affects the learning curve.

TABLE X
THE ANALYSIS QUANTIFIES COMPUTATIONAL COSTS OF EACH COMPONENT IN GOT-JEPA-L-378.

	Model Predictor		Reg/Cls	OccuSolver		Total
	Backbone	Predictor	Decoders	Point Tracker	Adapters	
Latency (ms)	23.82	4.39	2.30	9.43	1.39	41.34
Latency (%)	57.65	10.6	5.6	22.8	3.4	100
Trainable Parameter (M)	0	17.4	2.5	0	7.7	27.6
MACs (G)	125.5	13.9	1.8	182.8	1.1	325.1

Rows (3) and (4) demonstrate that copy-paste achieves larger performance gains when combined with JEPA, although both corruption strategies (*i.e.*, copy-paste or masking) yield improvements. During pre-training, copy-paste is applied in feature space to the current frame only, thereby simulating distractors and occlusion. All experiments in this evaluation are conducted at a resolution of 252 without using OccuSolver.

Learning Curve Analysis for Tracker Pre-Training from JEPA. Besides conducting an ablation study to analyze whether tracker pre-training through the JEPA architecture benefits the tracker, we also provide the learning curve. As indicated in Fig. 7, the tracker using "JEPA Pretrain," specifically validated in training stage 1, learns more effectively than when the JEPA pre-training strategy is not used.

ProjNet Usability Evaluation. We evaluate ProjNet in Tab. IX. Row (1) shows the baseline tracker, and row (2) adds the ProjNet tail to the baseline predictor. Row (3) removes ProjNet during both training and inference of GOT-JEPA, while row (4) includes it. ProjNet has minimal impact on the baseline tracker (rows (1) and (2)) due to its lightweight single-layer (1×1) convolution design. Row (3) confirms that JEPA-based predictor pre-training improves tracking over the baseline, and row (4) shows further gains when ProjNet is integrated. ProjNet enhances model prediction learning by projecting the student's representations into the teacher space, aligning corrupted and clean predictions.

E. Limitations and Future Work

While GOT-JEPA improves most attributes, as discussed in Section IV.C, its limitations are present in scenarios involving dense background clutter and fast motion, as evidenced by performance on the LaSOT and OTB-100 benchmarks. Furthermore, out-of-distribution data, such as those in AVisT, remains an open challenge for further improvement, especially under adverse imaging conditions (e.g., low-quality frames) and extreme target conditions (e.g., tiny objects moving rapidly). Looking forward, a promising direction to address background clutter is to incorporate 3D geometric cues, as most current trackers primarily rely on 2D semantic features. This can be achieved by augmenting training with auxiliary modalities (e.g., RGB-X data) or by leveraging recent advances in 3D geometry research to predict geometric representations from 2D images. Such geometry-aware cues may significantly enhance robustness in complex environments by more effectively separating the target from cluttered backgrounds.

TABLE XI
COMPARISON OF DIFFERENT QUERY POINTS FOR OCCUSOLVER AUC AS A METRIC ACROSS THREE DATASETS.

#Points	AVisT	LaSOT	OTB-100
64	61.2	72.9	70.3
128	61.7	73.3	70.6
256	61.8	73.3	70.4

TABLE XII
AN ABLATION STUDY ON WHETHER PREVENTING POINT SAMPLING FROM THE OCCLUDED FRAME CAN BENEFIT TRACKING.

First Frame Constraint	AVisT	LaSOT	OTB-100
-	63.55	75.03	73.24
✓	63.69	75.36	73.24

TABLE XIII
COMPARISON BETWEEN USING THE ORIGINAL POINT TRACKER FOR OCCUSOLVER AND OUR REFINEMENT STRATEGY ON THE AVIST.

Tracker	w/ Point Tracker	w/ LS-FineTune	w/ Obj. Prior	NPr	SUC	OP50
GOT-JEPA	✓			81.2	63.0	73.2
	✓	✓		81.3	63.3	73.3
	✓	✓	✓	81.9	63.7	73.7

V. COMPUTATIONAL COST ANALYSIS

Tab. X presents the per-frame runtime (ms), percentage of total time, trainable parameters, and multiply–accumulate operations (MACs) for each tracker component. With high-resolution inputs (378×378), the Model Predictor uses DINO features to estimate the tracking model, while the Adapters in OccuSolver refine the Point Tracker output, and the Reg/Cls decoders generate the final predictions. The primary computational bottleneck is the Backbone, which employs ViT-L at a large spatial resolution. The Point Tracker also incurs a notable cost, as it processes multiple query points. Since the cost scales linearly with the number of points [12], we fix the number of query points to 128 for efficiency, compared with dense tracking tasks that process millions of points (see Tab. XI).

Frame Sampling Strategy of OccuSolver: Fig. 9 examines the effect of the sampling step (N) between consecutive frames processed by OccuSolver. Since OccuSolver uses a fixed number of input frames, (N) defines a trade-off. A small (N) improves performance on moderate motion but weakens robustness to long occlusions, whereas a large (N) increases tracking failures under significant motion. Our results show (N=8) achieves the best balance.

To enhance stability, we add an inference constraint that prevents initialization on heavily occluded targets. If the first-frame visibility score drops below a threshold (e.g., lower than

Fig. 8. Visual comparisons of tracking results from raw annotations, GOT-JEPA, and the baseline tracker ToMP-L across diverse video sequences under adverse scenarios are presented. Rows 1 to 3 illustrate object tracking under occlusion, including partial and full occlusion, and the tracker's recovery after occlusion removal. Row 4 shows occlusion under distractor cases. Row 5 shows tracking with small targets. Row 6 presents tracking results for deformation cases. Best viewed when zoomed in; more detailed visual comparisons among trackers are provided in the video appendix.

85% visible points), initialization is skipped. During tracking, the input window follows a FIFO scheme; when a new frame is predicted occluded, the last unoccluded frame is duplicated. As shown in Tab. XII, this yields consistent gains (up to 0.3). We apply this constraint by default.

Refinement of Point Tracker. We provide an ablation study in Tab. XIII to quantify the impact of different point-tracker (CoTracker) variants in GOT-JEPA. Each row uses GOT-JEPA as the base tracker. The first setting uses only the pretrained point tracker and its predicted visibility, which is converted by our proposed mapping function for GOT feature refinement, without ladder-side fine-tuning or object priors. The second setting adds ladder-side fine-tuning of the point tracker, and the third setting further incorporates GOT-derived object priors. As shown in Tab. XIII, ladder-side fine-tuning of the point tracker and GOT-derived object priors yields consistent improvements, and the full setting achieves the best performance. We further clarify that CoTracker alone is not an occlusion solution for GOT. CoTracker is class-agnostic and outputs point trajectories without target identity and without target-specific query initialization, so it does not provide object-aware visibility on its own. In our framework, GOT-derived object priors and ladder-side fine-tuning make the point trajectories target-centric and align them with the GOT tracking feature space through our

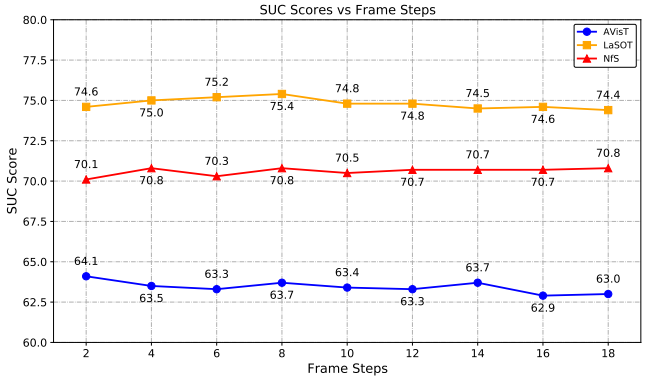


Fig. 9. An ablation study investigates the frame gap between consecutive sampled frames for OccuSolver during inference, using the SUC as the metric.

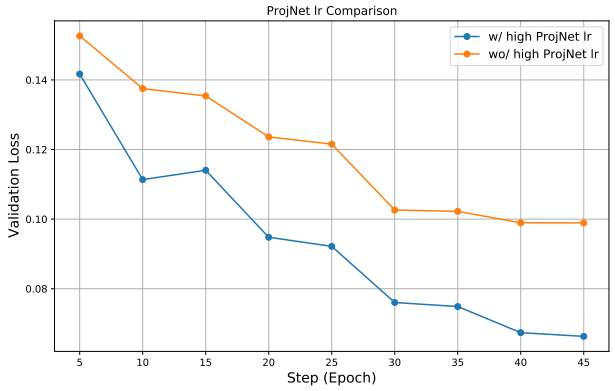


Fig. 10. Comparison of relative learning rates between ProjNet and other components during the JEPA training stage.

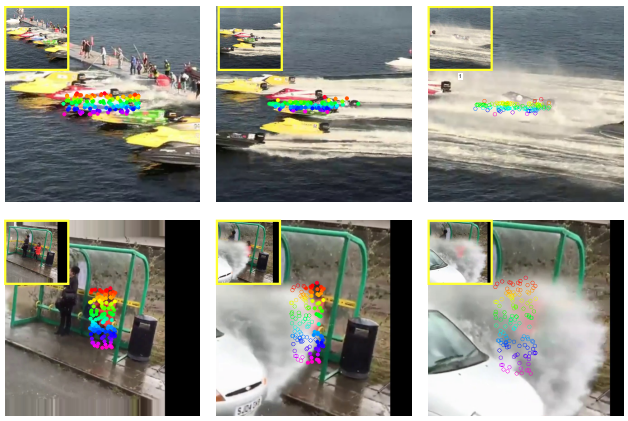


Fig. 11. Col 1: Initial frame and initial point sampling. Col 2: Points that are more object-aware and can handle partial occlusion. Col 3: Cases of full occlusion. The original input image is shown in the yellow box.

proposed mapping function, producing object-aware visibility cues that the tracker can directly exploit.

Relative Learning Rate between ProjNet and Other Components. In Fig. 10, we analyze the performance difference in the relative learning rate between ProjNet and other components (such as the Expander and Student model predictor). A higher learning rate for ProjNet enhances the model's learning effectiveness. The final learning rate for ProjNet is ten times that of the other components, starting at 1e-3 and 1e-4, respectively. This experiment was conducted with an image resolution of 252 during the model predictor pre-training stage, focusing on learning to predict tracking models. Similar phenomena can be observed in previous works with JEPA-like structure, such as DirectPred [134].

A. Visualization Results

We present visual comparisons among trackers in Fig. 8. Our tracker achieves more robust tracking under occlusion (rows 1 to 3) and superior discrimination and robustness against distractors, small targets, and deformation (rows 4 to 6). These advantages stem from our proposed model-based prediction learning and OccuSolver, which enhance the tracker's ability to handle occlusions and resist distractors.

We also present OccuSolver's output in Fig. 11, illustrating how randomly initialized points are refined to enhance object awareness or handle occlusions. Each column shows the original image alongside the point sampling, followed by the prediction result: Col 1 presents the initial frame and point sampling; Col 2 shows points that are more object-aware and handle partial occlusion; Col 3 presents cases of full occlusion. More qualitative results are in the supplementary material.

VI. Conclusion

In this work, we introduced GOT-JEPA, a tracking framework that enhances online model prediction and adaptation by advancing the JEPA from image feature prediction to the novel task of tracking model prediction. A teacher predictor generates pseudo-tracking models, while a student predictor learns to predict these models from corrupted current frames, given identical previous frame information. This design empowers the student model to both learn from diverse tracking concepts and account for potential variations, allowing it to predict an adapted and generalized model for the current frame. Equipped with OccuSolver, GOT-JEPA further enhances occlusion perception by generating precise, object-aware visibility states. This information not only improves occlusion handling but also produces higher-quality reference labels that incrementally refine and enhance subsequent model predictions, thereby boosting the tracker's overall robustness and adaptability in complex environments.

References

[1] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019. 1, 2, 4, 7
[2] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019. 1, 2
[3] S. Javed, M. Danelljan, F. S. Khan, M. H. Khan, and J. Matas, "Visual object tracking with discriminative filters and siamese networks: a survey and outlook," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2022. 1, 2, 4
[4] C. Mayer, M. Danelljan, G. Bhat, M. Paul, D. P. Paudel, F. Yu, and L. Van Gool, "Transforming model prediction for tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022. 1, 2, 3, 4, 5, 6, 7, 9, 10
[5] S.-F. Chen, J.-C. Chen, I.-H. Jhuo, and Y.-Y. Lin, "Improving visual object tracking through visual prompting," *IEEE Transactions on Multimedia (TMM)*, 2025. 1, 2, 3, 4, 5, 6, 7, 9, 10
[6] X. Wang, Z. Chen, J. Tang, B. Luo, Y. Wang, Y. Tian, and F. Wu, "Dynamic attention guided multi-trajectory analysis for single object tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2021. 2, 3
[7] Y. Bai, Z. Zhao, Y. Gong, and X. Wei, "Prompting autoregressive tracker where to look and how to describe," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024. 2, 3, 7
[8] C. Fan, H. Yu, Y. Huang, C. Shan, L. Wang, and C. Li, "Siamon: Siamese occlusion-aware network for visual tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2021. 2, 3, 7
[9] Y. Wu, X. Wang, X. Yang, M. Liu, D. Zeng, H. Ye, and S. Li, "Learning occlusion-robust vision transformers for real-time uav tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2025. 2, 3
[10] Y. Xue, T. Shen, G. Jin, L. Tan, N. Wang, L. Wang, and J. Gao, "Handling occlusion in uav visual tracking with query-guided redetection," *IEEE Trans. Instrum. Meas. (TIM)*, 2024. 2
[11] Y. LeCun, "A path towards autonomous machine intelligence," https://openreview.net/forum?id=BZ5a1r-kVsf, 2022. 2, 3, 4
[12] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht, "Cotracker: It is better to track together," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024. 2, 3, 5, 11
[13] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," *IEEE Trans. Image Process. (TIP)*, 2016. 2

[14] R. Yao, G. Lin, C. Shen, Y. Zhang, and Q. Shi, "Semantics-aware visual object tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2019. 2

[15] Y. Zhang, X. Gao, Z. Chen, H. Zhong, H. Xie, and C. Yan, "Mining spatial-temporal similarity for visual tracking," *IEEE Trans. Image Process. (TIP)*, 2020. 2

[16] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2011. 2

[17] Y. Huang, X. Li, Z. Zhou, Y. Wang, Z. He, and M.-H. Yang, "Rtracker: Recoverable tracking via pn tree structured memory," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024. 2

[18] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2015. 2

[19] Y. Yao, X. Wu, S. Shan, and W. Zuo, "Joint representation and truncated inference learning for correlation filter based tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018. 2

[20] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2017. 2

[21] A. Lukezic, T. Vojir, L. ˇCehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017. 2

[22] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010. 2

[23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012. 2

[24] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017. 2

[25] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016. 2, 6

[26] Y. Liang, H. Chen, Q. Wu, C. Xia, and J. Li, "Joint spatio-temporal similarity and discrimination learning for visual tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2025. 2

[27] W. Liu, Y. Song, D. Chen, S. He, Y. Yu, T. Yan, G. P. Hancke, and R. W. Lau, "Deformable object tracking with gated fusion," *IEEE Trans. Image Process. (TIP)*, 2019. 2

[28] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021. 2

[29] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017. 2

[30] D. Zhao, S. Kobayashi, J. Sacramento, and J. von Oswald, "Meta-learning via hypernetworks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020. 2, 4

[31] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018. 2

[32] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016. 2

[33] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016. 2

[34] J. Bao, K. Chen, X. Sun, L. Zhao, W. Diao, and M. Yan, "Siamthn: Siamese target highlight network for visual tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2025. 2

[35] L. Shen, X. Fan, and H. Li, "Overlapped trajectory-enhanced visual tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2024. 2

[36] J. Zhu, X. Chen, P. Zhang, X. Wang, D. Wang, W. Zhao, and H. Lu, "Srrt: Exploring search region regulation for visual object tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2024. 2

[37] D. Zhang, X. Xiao, Z. Zheng, Y. Jiang, and Y. Yang, "Probabilistic assignment with decoupled iou prediction for visual tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2024. 2

[38] Q. Gao, M. Yin, X. Wu, D. Liu, and Y. Bo, "Online multi-scale classification and global feature modulation for robust visual tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2024. 2

[39] M. Dong, K. Xu, X. Jiang, Z. Zhao, and T. Sun, "Feature-aware transferable adversarial attacks on visual object tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2025. 2

[40] B. Chen, P. Li, L. Bai, L. Qiao, Q. Shen, B. Li, W. Gan, W. Wu, and W. Ouyang, "Backbone is all your need: A simplified architecture for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022. 2

[41] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020. 2

[42] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020. 2

[43] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018. 2

[44] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "Siamcar: Siamese fully convolutional classification and regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020. 2

[45] P. Voigtlaender, J. Luiten, P. H. Torr, and B. Leibe, "Siam r-cnn: Visual tracking by re-detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020. 2

[46] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable siamese attention networks for visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020. 2

[47] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020. 2

[48] Z. Fu, Z. Fu, Q. Liu, W. Cai, and Y. Wang, "Sparsett: Visual tracking with sparse transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022. 2, 9

[49] F. Tang and Q. Ling, "Ranking-based siamese visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022. 2

[50] Z. Pi, W. Wan, C. Sun, C. Gao, N. Sang, and C. Li, "Hierarchical feature embedding for visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022. 2

[51] K. Zhao, D. Zhao, L. Xiao, Y. Nie, Y. Huang, and Y. Zhang, "Idstt: Iterative dual-sample-teacher for semi-supervised visual object tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2025. 2, 3

[52] Y. Chen, G. Jia, Y. Zha, P. Zhang, and Y. Zhang, "Linr: A plug-and-play local implicit neural representation module for visual object tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2025. 2

[53] X. Hu, B. Zhong, Q. Liang, S. Zhang, N. Li, X. Li, and R. Ji, "Transformer tracking via frequency fusion," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2024. 2

[54] L. Chen, L. Gao, Y. Jiang, Y. Li, G. He, and J. Ning, "Local-global self-attention for transformer-based object tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2025. 2

[55] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021. 3, 9

[56] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021. 3, 9

[57] B. Ye, H. Chang, B. Ma, S. Shan, and X. Chen, "Joint feature learning and relation modeling for tracking: A one-stream framework," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022. 3, 7, 8

[58] S. Gao, C. Zhou, and J. Zhang, "Generalized relation modeling for transformer tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023. 3, 7, 8, 9

[59] Y. Cai, J. Liu, J. Tang, and G. Wu, "Robust object modeling for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023. 3, 7, 9

[60] M. Guo, Z. Zhang, H. Fan, L. Jing, Y. Lyu, B. Li, and W. Hu, "Learning target-aware representation for visual tracking via informative interactions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022. 3

[61] S. Gao, C. Zhou, C. Ma, X. Wang, and J. Yuan, "Aiatrack: Attention in attention for transformer visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022. 3

[62] K. He, C. Zhang, S. Xie, Z. Li, and Z. Wang, "Target-aware tracking with long-term context attention," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2023. 3

[63] X. Zhou, P. Guo, L. Hong, J. Li, W. Zhang, W. Ge, and W. Zhang, "Reading relevant feature from global representation memory for visual object tracking," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023. 3

[64] X. Li, Y. Huang, Z. He, Y. Wang, H. Lu, and M.-H. Yang, "Citetracker: Correlating image and text for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023. 3

[65] G. Zeng, B. Zeng, Q. Wei, H. Hu, and H. Zhang, "Visual object tracking with mutual affinity aligned to human intuition," *IEEE Trans. Multimedia (TMM)*, 2024. 3

[66] J. Zhang, Z. Li, R. Wei, and Y. Wang, "Augment one with others: Generalizing to unforeseen variations for visual tracking," *IEEE Trans. Multimedia (TMM)*, 2025. 3

[67] B. Yan, H. Zhao, D. Wang, H. Lu, and X. Yang, "'skimming-perusal'tracking: a framework for real-time and robust long-term tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019. 3

[68] Z. Shao, Y. Hu, J. Liu, B. Fan, and H. Liu, "Group orthogonal low-rank adaptation for rgb-t tracking," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2026. 3

[69] X. Chen, H. Peng, D. Wang, H. Lu, and H. Hu, "Seqtrack: Sequence to sequence learning for visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023. 3, 6, 7, 8, 9

[70] X. Wei, Y. Bai, Y. Zheng, D. Shi, and Y. Gong, "Autoregressive visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023. 3

[71] X. Jinxia, Z. Bineng, M. Zhiyi, Z. Shengping, S. Liangtao, S. Shuxiang, and J. Rongrong, "Autoregressive queries for adaptive tracking with spatio-temporal transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024. 3

[72] L. Shi, B. Zhong, Q. Liang, N. Li, S. Zhang, and X. Li, "Explicit visual prompts for visual object tracking," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2024. 3, 9

[73] W. Cai, Q. Liu, and Y. Wang, "Hiptrack: Visual tracking with historical prompts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024. 3, 7, 9

[74] Z. Song, R. Luo, J. Yu, Y.-P. P. Chen, and W. Yang, "Compact transformer tracker with correlative masked modeling," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2023. 3, 8, 9

[75] Q. Wu, T. Yang, Z. Liu, B. Wu, Y. Shan, and A. B. Chan, "Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023. 3

[76] H. Zhao, D. Wang, and H. Lu, "Representation learning for visual object tracking by masked appearance transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023. 3

[77] X. Huang, D. Miao, H. Wang, Y. Wang, and X. Li, "Context-guided black-box attack for visual tracking," *IEEE Trans. Multimedia (TMM)*, 2024. 3

[78] Z. Zhang, L. Xu, D. Peng, H. Rahmani, and J. Liu, "Diff-tracker: Text-to-image diffusion models are unsupervised trackers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024. 3

[79] F. Xie, Z. Wang, and C. Ma, "Diffusiontrack: Point set diffusion model for visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024. 3, 8

[80] L. Huang, X. Zhao, and K. Huang, "Globaltrack: A simple and strong baseline for long-term tracking," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020. 3

[81] C. Doersch, A. Gupta, L. Markeeva, A. Recasens, L. Smaira, Y. Aytar, J. Carreira, A. Zisserman, and Y. Yang, "Tap-vid: A benchmark for tracking any point in a video," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022. 3

[82] A. W. Harley, Z. Fang, and K. Fragkiadaki, "Particle video revisited: Tracking through occlusions using point trajectories," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022. 3

[83] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman, "Tapir: Tracking any point with per-frame initialization and temporal refinement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023. 3

[84] Q. Wang, Y.-Y. Chang, R. Cai, Z. Li, B. Hariharan, A. Holynski, and N. Snavely, "Tracking everything everywhere all at once," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023. 3

[85] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023. 3

[86] A. Bardes, Q. Garrido, J. Ponce, X. Chen, M. Rabbat, Y. LeCun, M. Assran, and N. Ballas, "Revisiting feature prediction for learning visual representations from video," *Trans. Mach. Learn. Res. (TMLR)*, 2024. 3

[87] M. Abdelfattah and A. Alahi, "S-jepa: A joint embedding predictive architecture for skeletal action recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024. 3

[88] Z. Dong, R. Li, Y. Wu, T. T. Nguyen, J. S. X. Chong, F. Ji, N. R. J. Tong, C. L. H. Chen, and J. H. Zhou, "Brain-jepa: Brain dynamics foundation model with gradient positioning and spatiotemporal masking," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2024. 3

[89] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019. 4

[90] X. Li, C. Huang, C.-L. Li, E. Malach, J. Susskind, V. Thilak, and E. Littwin, "Rethinking jepa: Compute-efficient video ssl with frozen teachers," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2026. 4

[91] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born-again neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018. 4, 5

[92] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *Trans. Mach. Learn. Res. (TMLR)*, 2023. 4, 7, 10

[93] D.-N. Grigore, M.-I. Georgescu, J. A. Justo, T. Johansen, A. I. Ionescu, and R. T. Ionescu, "Weight copy and low-rank adaptation for few-shot distillation of vision transformers," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. WACV)*, 2025. 4, 5

[94] D. Ha, A. M. Dai, and Q. V. Le, "Hypernetworks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017. 4

[95] S. Schug, S. Kobayashi, Y. Akram, J. Sacramento, and R. Pascanu, "Attention as a hypernetwork," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025. 4

[96] A. Bardes, J. Ponce, and Y. LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022. 4, 5

[97] K. Ranasinghe, M. Naseer, M. Hayat, S. Khan, and F. S. Khan, "Orthogonal projection loss," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021. 5

[98] Z. Zhang and M. Sabuncu, "Self-distillation as instance-specific label smoothing," in *Adv. Neural Inform. Process. Syst. (NIPS)*, 2020. 5

[99] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021. 5

[100] L. Schmarje, V. Grossmann, C. Zelenka, S. Dippel, R. Kiko, M. Oszust, M. Pastell, J. Stracke, A. Valros, N. Volkmann *et al.*, "Is one annotation enough? a data-centric image classification benchmark for noisy and ambiguous label estimation," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022. 5

[101] Y.-L. Sung, J. Cho, and M. Bansal, "Lst: Ladder side-tuning for parameter and memory efficient transfer learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022. 5

[102] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020. 5

[103] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 24 261–24 272. 5

[104] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2019. 6, 7

[105] Y. Ma, Y. Tang, W. Yang, T. Zhang, X. Zhou, and F. Wu, "Unisot: a unified framework for multi-modality single object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2026. 7

[106] C.-Y. Yang, H.-W. Huang, W. Chai, Z. Jiang, and J.-N. Hwang, "Samurai: Motion-aware memory for training-free visual object tracking with sam 2," *IEEE Transactions on Image Processing*, 2026. 7

[107] J. Li, X. Yuan, H. Qin, Y. Wang, X. Liu, and T. Xu, "Cvt-track: Concentrating on valid tokens for one-stream tracking," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2025. 7

[108] W. Wang, M. Lv, L. Zhu, T. Han, Y. Zhang, and Y. Li, "Siamese visual tracking with multi-parallel interactive transformers," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 7

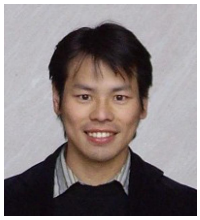
[109] C. Xue, B. Zhong, Q. Liang, H. Xia, and S. Song, "Unifying motion and appearance cues for visual tracking via shared queries," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2025. 7

[110] Y. Han, M. Cai, J. Wu, Z. Bai, T. Zhuo, H. Zhang, and Y. Zhang, "Visual object tracking with multi-frame distractor suppression," *IEEE Trans. Circ. Syst. Video Tech. (TCSVT)*, 2025. 7

[111] J. Xiong and Q. Ling, "Mask-guided siamese tracking with a frequency-spatial hybrid network," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 7
[112] Y. Ma, Q. Yu, W. Yang, T. Zhang, and J. Zhang, "Learning discriminative features for visual tracking via scenario decoupling," *Int. J. Comput. Vis. (IJCV)*, 2025. 7
[113] F. Xie, W. Yang, C. Wang, L. Chu, Y. Cao, C. Ma, and W. Zeng, "Correlation-embedded transformer tracking: A single-branch framework," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2024. 7
[114] L. Lin, H. Fan, Z. Zhang, Y. Wang, Y. Xu, and H. Ling, "Tracking meets lora: Faster training, larger model, stronger performance," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024. 6, 7
[115] M. Noman, W. A. Ghallabi, D. Najiha, C. Mayer, A. Dudhane, M. Danelljan, H. Cholakkal, S. Khan, L. Van Gool, and F. S. Khan, "Avist: A benchmark for visual object tracking in adverse visibility," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2022. 6, 10
[116] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2017. 6
[117] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2015. 6
[118] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019. 6
[119] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, M. Danelljan, A. Lukežič *et al.*, "The tenth visual object tracking vot2022 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022. 6, 7
[120] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021. 7
[121] X. Yang, C. Zhao, J. Yang, Y. Song, and Y. Zhao, "Negative-driven training pipeline for siamese visual tracking," *IEEE Trans. Multimedia (TMM)*, 2024. 7
[122] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019. 7
[123] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019. 7
[124] J. Xie, B. Zhong, Q. Liang, N. Li, Z. Mo, and S. Song, "Robust tracking via mamba-based context-aware token learning," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2025. 7
[125] X. Li, B. Zhong, Q. Liang, G. Li, Z. Mo, and S. Song, "Mambalct: Boosting tracking via long-term context state space model," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2025. 7
[126] Y. Ma, Y. Tang, W. Yang, T. Zhang, J. Zhang, and M. Kang, "Unifying visual and vision-language tracking via contrastive learning," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2024. 8, 9
[127] Z. Song, J. Yu, Y.-P. P. Chen, and W. Yang, "Transformer tracking with cyclic shifting window attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022. 8, 9
[128] Y. Zheng, B. Zhong, Q. Liang, Z. Mo, S. Zhang, and X. Li, "Odtrack: Online dense temporal token learning for visual tracking," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2024. 8, 9
[129] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018. 9
[130] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016. 9
[131] Y. Kou, J. Gao, B. Li, G. Wang, W. Hu, Y. Wang, and L. Li, "Zoomtrack: Target-aware non-uniform resizing for efficient visual tracking," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023. 9
[132] Y. Cui, C. Jiang, L. Wang, and G. Wu, "Mixformer: End-to-end tracking with iterative mixed attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022. 9
[133] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021. 9
[134] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021. 13

Shih-Fang Chen is a Ph.D. candidate in the Department of Computer Science at National Yang Ming Chiao Tung University, Taiwan. His research interests primarily focus on computer vision and machine learning. He actively contributes to the research field as a reviewer for several leading international journals and conferences, including IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), IEEE Transactions on Multimedia (TMM), and the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Jun-Cheng Chen (Member, IEEE) is an Associate Research Fellow at the Research Center for Information Technology Innovation (CITI), Academia Sinica. He joined CITI as an assistant research fellow in 2019. He received the BS and MS degrees advised by Prof. Ja-Ling Wu in Computer Science and Information Engineering from National Taiwan University, Taiwan (R.O.C), in 2004 and 2006, respectively, where he received the PhD degree advised by Prof. Rama Chellappa in Computer Science from University of Maryland, College Park, USA, in 2016. From 2017 to 2019, he was a postdoctoral research fellow at the University of Maryland Institute for Advanced Computer Studies. His research interests include computer vision, machine learning, deep learning and their applications to biometrics, such as face recognition/facial analytics, activity recognition/detection in the visual surveillance domain, etc. His works have been recognized in prestigious journals and conferences in the field, including PNAS, TBIOM, CVPR, ICCV, ECCV, ICLR, WACV, etc. He was a recipient of the ACM Multimedia Best Technical Full Paper Award in 2006, APSIPA ASC Best Paper Award in 2023, and IEEE CE Magazine Best Paper Award in 2025.



I-Hong Jhuo (Member, IEEE) is a Principal Applied Scientist at Microsoft AI. He actively contributes to the development of innovative technologies for computer vision, information retrieval and large-scale data systems. His research interests include computer vision, information retrieval and artificial intelligence. His work has been recognized in international competitions, including the ACM Multimedia Grand Challenge 2012. He also contributed to the design of a top-performing video analytics system for the NIST TRECVID Multimedia Event Detection (MED) evaluation in collaboration with Columbia University, New York, NY, USA.

Yen-Yu Lin (Senior Member, IEEE) received the B.B.A. degree in Information Management, and the M.S. and Ph.D. degrees in Computer Science and Information Engineering from National Taiwan University, Taipei, Taiwan, in 2001, 2003, and 2010, respectively. He is currently a Distinguished Professor with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan. His research interests include computer vision, machine learning, and artificial intelligence. He has served as an Area Chair for ICCV, CVPR, and ECCV. He is currently an Associate Editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) and the International Journal of Computer Vision (IJCV).