

LETTER OPEN ACCESS

VT-Surf: Visual Tracking with Switching Dynamics under Time-Series Forecasting

Seonghak Lee¹  | Jisoo Park² | Radu Timofte³ | Junseok Kwon^{1,2} 

¹School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea | ²Graduate School of Artificial Intelligence, Chung-Ang University, Seoul, South Korea | ³Computer Vision Lab, CAIDAS and IFI, University of Würzburg, Würzburg, Germany

Correspondence: Junseok Kwon (jskwon@cau.ac.kr)

Received: 11 August 2025 | **Revised:** 9 November 2025 | **Accepted:** 14 November 2025

ABSTRACT

In this paper, we reconceptualize visual tracking as a multivariate time-series forecasting (MTSF) problem. Specifically, the goal of visual tracking—predicting the target's state over time, including its (x, y) center coordinates and scale—can be naturally framed as forecasting future states from a sequence of past observations. Viewed through this lens, visual tracking aligns with the challenges of MTSF, where the objective is to capture complex temporal dependencies among multiple variables. However, applying MTSF to visual tracking introduces new difficulties due to the inherently intricate nature of object motion, which often involves abrupt and nonlinear variations in direction, velocity, and behavioral patterns. To address these complexities, we propose a principled approach that models the dynamic nature of target motion using a regime-switching framework. This method employs an underlying Markov jump process (MJP) to govern transitions between multiple latent motion patterns, each characterized by its own stochastic differential equation (SDE). By doing so, our model adapts to diverse temporal dynamics in a data-driven manner, enabling robust and precise prediction of future target states. Experimental results demonstrate that our method outperforms state-of-the-art visual tracking approaches, particularly in scenarios where target objects exhibit diverse and dynamic motion patterns over time.

1 | Introduction

Visual tracking is a core technique widely used to predict the trajectories of target objects and serves as a foundation for various computer vision applications, including medical imaging, autonomous driving, surveillance, and robotics. Self-SDCT [1] introduced a self-supervised learning-based visual tracker designed within a deep correlation framework. ALT [2] proposed an active learning approach for deep visual tracking that selects and annotates unlabeled samples to train deep CNN models. In many studies, visual tracking has been formulated stochastically within the framework of state forecasting, where the goal is to accurately estimate future states of a target based on its historical state information. For example, PrDiMP [3] formulated visual tracking as probabilistic regression, while EWLCFDP [4] employed Markov Chain Monte Carlo to handle uncertainty

in visual tracking. Within this context, we propose that visual tracking can be effectively addressed using mathematical tools developed for time-series data forecasting.

Multivariate time series forecasting (MTSF) has gained increasing importance across a wide range of domains, including finance, climate science, transportation, and energy. MTSF predicts future values of multiple interrelated variables based on their past behaviors, where multivariate refers to interacting signals (e.g., (x, y) , and scale) and time-series forecasting involves predicting future values over time from historical data. A critical challenge in MTSF is effectively capturing temporal dependencies among multiple interacting variables. To address this, models based on convolutional neural networks (CNNs) [5] and recurrent neural networks (RNNs) [6] have been proposed. However, these architectures often struggle with modeling long-term dependencies in

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2025 The Author(s). *Electronics Letters* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

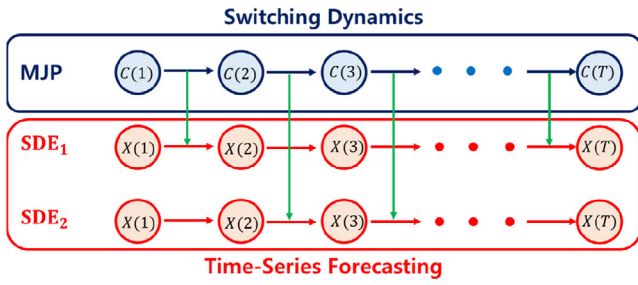


FIGURE 1 | Pipeline of the proposed visual tracking framework (VT-Surf). Blue: Markov jump process for switching among multiple SDEs. Red: SDE process for time-series forecasting. Green: switching SDEs.

time series data. Recently, Transformer-based architectures [7] have emerged as a more effective alternative, leveraging self-attention mechanisms to better capture long-range temporal relationships.

When applied to visual tracking, MTSF becomes more challenging due to the increased complexity of target object motions, which can exhibit significant variations in direction, speed, and behavioral patterns. To address this challenge, we adopt a methodology capable of handling diverse time series dynamics by modeling switching patterns in the data [8], governed by an underlying Markov jump process (MJP) [9]. In this framework, distinct time series dynamics are modeled using different stochastic differential equations (SDEs), with a discrete MJP governing the switching between the appropriate SDEs. By leveraging this approach, the proposed tracker can effectively accommodate variations in direction, speed, and behavioral patterns, enabling accurate time series forecasting of future target states based on the history of past observations. The contributions of our method are as follows:

- We propose a novel visual tracking framework formulated as a multivariate time-series forecasting problem.
- We introduce a Markov jump process to govern transitions among diverse latent motion patterns, each modeled by a distinct stochastic differential equation (SDE).

Figure 1 illustrates the pipeline of the proposed visual tracking framework.

2 | Visual Tracking via Time-Series Forecasting

In visual tracking, the target state at time t , denoted by $X_t = \{x_t, y_t, s_t\}$, represents the spatial and scale configurations of the object being tracked, where: x_t, y_t denote horizontal and vertical positions of the target, respectively, and s_t is the scale of the target. Then, the goal of visual tracking is to estimate the current state X_t as accurately as possible using all available information up to time t . This includes previous target states $\mathbf{X}_{1:t-1} = \{X_1, \dots, X_{t-1}\}$ and the corresponding observations $\mathbf{Z}_{1:t} = \{Z_1, \dots, Z_t\}$. This objective is typically formulated using the joint probability of the target state sequence and the corresponding observation sequence: $p(X_{1:t}, Z_{1:t}) = \prod_{\tau=1}^t p(Z_\tau | X_{1:\tau}, Z_{1:\tau-1}) \times p(X_\tau | X_{1:\tau-1}, Z_{1:\tau-1})$, where the current state X_τ typically depends only on the past target states $X_{1:\tau-1}$, and is

conditionally independent of the past observations $Z_{1:\tau-1}$. Then, the transition model $p(X_\tau | X_{1:\tau-1}, Z_{1:\tau-1})$ can be simplified as $p(X_\tau | X_{1:\tau-1})$. Similarly, the current observation Z_τ is typically assumed to depend only on the current target state X_τ , and to be conditionally independent of the past observations $Z_{1:\tau-1}$ and past states $X_{1:\tau-1}$. Under this assumption, the observation model $p(Z_\tau | X_{1:\tau}, Z_{1:\tau-1})$ can be simplified as $p(Z_\tau | X_\tau)$. Consequently, the joint probability $p(X_{1:t}, Z_{1:t})$ can be rewritten as $p(X_{1:t}, Z_{1:t}) = \prod_{\tau=1}^t p(Z_\tau | X_\tau) \times p(X_\tau | X_{1:\tau-1})$, where the terms $p(Z_\tau | X_\tau)$ and $p(X_\tau | X_{1:\tau-1})$ denote the likelihood and the state transition probability at time τ , respectively. The likelihood represents the similarity between the observed feature $Z_t(X_t)$ (i.e., the feature extracted from the region specified by X_t) and the reference feature Z_{ref} , typically measured using a distance metric D : $p(Z_t | X_t) = e^{-D(Z_t(X_t), Z_{ref})}$, where we adopt deep features used in [10] as the observed features and the correlation used in [11] as the distance metric.

3 | Continuous-Time State Transition Model

In particular, unlike conventional visual tracking methods, we design the state transition probability $p(X_\tau | X_{1:\tau-1})$ using mathematical tools from MTSF. To this end, we introduce a diffusion process [12] formulated via a SDE for $X(t)$, which models the future state X_τ conditioned on the past states $X_{1:\tau-1}$, that is, $p(X_\tau | X_{1:\tau-1})$:

$$dX(t) = \mu(X(t), t) dt + \sigma(X(t), t) dW(t), \text{ where:} \quad (1)$$

- $\mu(X(t), t)$ is the *drift* term, representing the deterministic trend or average rate of change of the process.
- $\sigma(X(t), t)$ is the *diffusion* term, controlling the magnitude of random fluctuations.
- $W(t)$ is a *Wiener process*, a continuous-time stochastic process with independent, normally distributed increments and $W(0) = 0$.

4 | Markov Jump Switching Process

When applied to visual tracking, MTSF becomes more challenging due to the increased complexity of target object motions, which may exhibit substantial variations in direction, speed, and behavioral patterns. To address this challenge, we adopt a methodology that accommodates diverse time-series dynamics by modeling switching patterns in the data [8], governed by an underlying MJP. Specifically, the MJP is implemented as a continuous-time switching process $\{C(t)\}_{t \geq 0}$, which selects among multiple SDEs to accurately capture MTSF variations in direction, speed, and behavioral patterns. We model $C(t)$ as a time-continuous Markov process, specifying its transition rate function, and exit rates:

- The transition rate function $q_{ij}(t)$ governs the switching of SDEs from the i -th SDE to the j -th SDE: $q_{ij}(t) := \lim_{\delta \rightarrow 0} \delta^{-1} p(C(t + \delta) = j | C(t) = i)$. When $i = \text{SDE}_i$ and $j = \text{SDE}_j$, the transition rate function facilitates the change from SDE_i to SDE_j because the latter provides a better

representation of future variations in direction, speed, and behavioral patterns.

- The exit rate is defined as $q_{ii}(t) := -\sum_{j \in \mathcal{I} \setminus i} q_{ij}(t)$, where $\mathcal{I}$ denotes the set of all indices of the SDEs.

5 | Switching Multiple State Transition Models

Incorporating the aforementioned continuous-time switching process $C(t)$ into a diffusion process $dX(t)$ in (2) yields a switching SDE:

$$dX(t) = \mu(C(t), X(t), t) dt + \sigma(C(t), X(t), t) dW(t), \text{ where: } (2)$$

- $C(t)$ is realized as a discrete value c , referred to as the mode, and can be simulated using the Doob–Gillespie algorithm [13].
- $\mu(C(t), X(t), t)$ is implemented via a mode (c)-dependent linear time (t)-invariant stochastic system [8], that is, $\mu(C(t), X(t), t) = \mu(c, x)$, where the realization of $X(t)$, denoted by x , is referred to as the state. In particular, $\mu(c, x)$ can be parameterized using the affine drift function specific to each mode c , that is, $\mu(c, x) = A(c)x + b(c)$, where $A(c) \in \mathbb{R}^{n \times n}$ is the state transition matrix for mode c , $b(c) \in \mathbb{R}^n$ is the constant drift vector for mode c , and n denotes the dimension of the state.
- $\sigma(C(t), X(t), t)$ is assumed to be a time (t)-homogeneous and state (x)-independent dispersion, that is, $\sigma(C(t), X(t), t) = \sigma(c)$, where $\sigma(c) \in \mathbb{R}^{n \times m}$ and m denotes the dimension of the Wiener process $W(t)$.

Depending on the motion dynamics, the MJP utilizes three to five Markov modes, exhibiting transition rates in the range of 0.12–0.33 across datasets (Average LaSOT: 4.32 (0.27), UAV123: 4.14 (0.29), OTB100: 3.02 (0.12), NFS: 5.12 (0.31), and VOT2020: 5.32 (0.33)). For training, $A(c)$ and $b(c)$ describe the deterministic trend of the target's motion for each mode c . They are optimized to capture deterministic motion trends in the training data. Concretely, they are learned by minimizing the difference between predicted and observed state transitions based on the maximum-likelihood estimation process. The diffusion term $\sigma(c)$ is estimated from the covariance of the residuals, that is, the variability that $A(c)$ and $b(c)$ cannot explain. $C(t)$ is simulated using the Doob–Gillespie algorithm, allowing data-driven learning of switching behavior between modes.

6 | Experiments

We compared our visual tracker called **VT-Surf** with several state-of-the-art trackers. For evaluations, we used five benchmark datasets: OTB100 [14], LaSOT [15], UAV123 [16], VOT2020 [17], and NFS [18]. We evaluated performance using seven metrics: precision plot, normalized precision, success plot, area under the curve (AUC), accuracy, robustness, and expected average overlap (EAO).

All experiments were conducted on a desktop computer equipped with a 3.60 GHz Intel Core i7 CPU and a GeForce Titan X GPU. Our visual tracking method was implemented in PyTorch.

TABLE 1 | Comparison on LaSOT. The **best**, **second best**, and **third best** results were highlighted.

	Precision $\uparrow$	Normalized precision $\uparrow$	AUC $\uparrow$
Stark [19]	72.2	76.9	67.1
TransT [20]	69.0	73.8	64.9
STMTrack [21]	63.3	69.3	60.6
PrDiMP [3]	60.8	68.8	59.8
SAOT [22]	—	70.8	61.6
CIA [23]	71.5	—	67.6
ToMP [24]	73.5	79.2	68.5
CiteTracker [25]	—	78.6	69.7
DiffusionTrack [26]	76.7	79.8	70.8
One Tracker [27]	76.5	79.9	70.5
VT-Surf	73.6	79.9	69.2

6.1 | Tracking Results

Table 1 presents the visual tracking results of state-of-the-art methods on the LaSOT dataset. Three evaluation metrics were employed in this experiment: precision, normalized precision, and AUC, where higher values indicate superior tracking accuracy. The proposed VT-Surf achieved the highest performance in both the precision and normalized precision metrics, while ToMP ranked second. CIA, Stark, KeepTrack, and SuperDiMP also demonstrated competitive tracking performance. The LaSOT dataset is characterized by long-term variations in target motion, including changes in speed and direction, leading to substantial differences between the initial and subsequent states. To cope with these challenges, the proposed method adaptively switches between time-series prediction models and continuously estimates the target's dynamically varying states. We further compared our method with recent diffusion-based trackers [26] and vision foundation-based trackers [27]. Although their tracking accuracy is comparable to ours, both DiffusionTrack and OneTracker require significantly longer training times due to the use of diffusion and large-scale foundation models.

The temporal analysis investigates the mode-switching behavior of the MJP over time across different datasets. In the LaSOT dataset, the temporal properties (e.g., Markov modes) vary notably when fast-moving targets appear in drone-1, nonlinear trajectories are observed in dog-3, or when combined object and camera motion occurs in bus-19. The efficiency assessment, as summarized in 4, demonstrates that VT-Surf operates in real time (97 FPS) while maintaining competitive tracking accuracy. We also conducted pairwise Wilcoxon signed-rank tests on per-sequence scores (success AUC and precision) for the LaSOT dataset. VT-Surf showed a median success-AUC improvement of +1.2 percentage points [0.7, 1.8] compared with ToMP. A Wilcoxon signed-rank test yielded $p = 0.003$, and Cliff's $\delta = 0.18$ (small-to-medium effect size). A permutation test produced $p = 0.004$, and the win rate was 61%.

Table 2 reports the visual tracking results in terms of success AUC on three benchmark datasets: UAV123, OTB100, and NFS.

TABLE 2 | Comparison on UAV123/ OTB100/ NFS in success AUC.

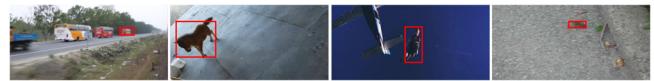
	UAV123 [16]	OTB100 [14]	NFS [18]
Stark [19]	68.2	68.1	66.2
TransT [20]	69.1	69.4	65.7
STMTrack [21]	64.7	71.9	—
TrSiam [28]	70.3	67.4	65.8
PrDiMP [3]	68.0	69.6	63.5
CIA [23]	71.3	68.9	66.7
SiamRPN++ [29]	69.6	61.3	50.2
DiMP [30]	65.3	68.4	62.0
EWLCFDP [4]	—	68.6	—
ToMP [24]	69.0	70.1	66.9
CRACK [31]	66.4	72.6	62.5
VT-Surf	70.5	72.6	67.1

TABLE 3 | Comparison on VOT2020 in terms of EAO score.

	Accuracy ↑	Robustness ↑	EAO ↑
Stark [19]	0.478	0.799	0.308
SuperDiMP [3, 30]	0.477	0.728	0.30
CIA [23]	0.481	0.782	0.309
DiMP [30]	0.457	0.734	0.274
UPDT [32]	0.465	0.755	0.278
DPMT [33]	0.492	0.745	0.303
TRAT [34]	0.464	0.744	0.280
ToMP [24]	0.453	0.814	0.309
VT-Surf	0.491	0.816	0.313

In this experiment, the compared methods were evaluated using the AUC metric. VT-Surf achieved state-of-the-art performance across all datasets. The NFS dataset, in particular, was recorded at high frame rates and contains numerous targets undergoing substantial state variations over time. Despite this challenge, the proposed method maintained robust tracking performance. This is attributed to the continuous-time state transition model, which accurately estimates the target state based on its past trajectory, and the Markov jump switching process, which effectively accounts for large discrepancies between current and previous motion patterns. While ToMP achieved the best individual performance on the NFS dataset and CRACK achieved the highest performance on the OTB100 dataset, VT-Surf consistently delivered the best overall tracking results across all benchmarks.

Table 3 presents the quantitative results on the VOT2020 dataset. In this experiment, several visual tracking methods were evaluated using three metrics: accuracy, robustness, and EAO. The VOT2020 dataset consists of 60 challenging video sequences in which target objects frequently lose their trajectories due to various obstacles and complex real-world conditions. Among these metrics, robustness is particularly important for assessing the ability of a tracker to maintain performance under such adverse

**FIGURE 2** | Qualitative results of the proposed method on the LaSOT dataset.**TABLE 4** | Computational efficiency. The computational efficiency was evaluated in terms of frames per second (FPS).

	Stark	TransT	ToMP	SiamRPN++	DiMP	Ours
FPS	66	8	6	10	127	97

TABLE 5 | Ablation study on VOT2020 with and without MJP.

	Accuracy ↑	Robustness ↑	EAO ↑
SDE (time-series forecasting)	0.487	0.809	0.310
SDE+MJP	0.491	0.816	0.313

conditions. As shown in Table 3, our tracker achieved the best performance in most metrics and attained the highest EAO score. DPMT, ToMP, and CIA also demonstrated competitive, state-of-the-art results on this dataset. Figure 2 shows the qualitative results on the LaSOT dataset. Although *Bus-19*, *Horse-1*, *Person-10*, and *Mouse-9* contain severe blurring, illumination changes, deformations, and background clutter, the proposed method successfully tracked the targets.

Table 4 reports the computational efficiency in terms of FPS. Our method runs in real time, because additional MJP components introduce only a negligible computational burden to visual tracking. In Table 5, we conducted an ablation study evaluating our method with and without MJP. The results clearly demonstrate that both core components, SDE and MJP, play pivotal roles in achieving robust and accurate visual tracking. Specifically, removing the MJP leads to noticeable degradation in performance, highlighting its importance in modeling abrupt motion transitions and dynamic state changes of the target.

7 | Conclusion

In this paper, we formulate visual tracking as a MTSF problem, where the target's state is predicted from past observations. To address the abrupt and nonlinear variations in object motion, we introduce a switching framework driven by a MJP, which transitions between latent motion patterns modeled by distinct SDEs. This adaptive approach effectively captures diverse temporal dynamics, enabling robust and accurate state prediction. Experimental results show that our method surpasses state-of-the-art trackers, particularly in scenarios with complex and highly dynamic motion.

Author Contributions

Seonghak Lee: conceptualization, methodology, writing – original draft. **Isosoo Park:** conceptualization, methodology, validation. **Radu Timofte:**

writing – review and editing. **Junseok Kwon:** writing – review and editing, supervision.

Funding

This research was supported in part by the Chung-Ang University Research Grants in 2024; in part by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korean government (MSIT) (RS-2021-II211341, Artificial Intelligence Graduate School Program [Chung-Ang University]), and in part by the Humboldt Foundation.

Conflicts of Interest

The authors declare no conflicts of interests.

Data Availability Statement

Data available on request from the authors.

References

1. D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, “Self-Supervised Deep Correlation Tracking,” *IEEE Transactions on Image Processing* 30 (2021): 976–985.
2. D. Yuan, X. Chang, Q. Liu, et al., “Active Learning for Deep Visual Tracking,” *IEEE Transactions on Neural Networks and Learning Systems* 35, no. 10 (2024): 13284–13296.
3. M. Danelljan, L. V. Gool, and R. Timofte, “Probabilistic Regression for Visual Tracking,” in *Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2020).
4. H. Zhang, G. Nie, J. Chen, J. Zhang, and G. Yang, “Uncertain Motion Tracking Combined Markov Chain Monte Carlo and Correlation Filters,” *IEEE Access* 7 (2019): 167076–167088.
5. G. Lai, W.-C. Chang, Y. Yang, and H. Liu, “Modeling Long-and Short-Term Temporal Patterns with Deep Neural Networks,” in *ACM SIGIR* (ACM, 2018).
6. D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, “Deepar: Probabilistic Forecasting with Autoregressive Recurrent Networks,” *International Journal of Forecasting* 36, no. 3 (2020): 1181–1191.
7. A. Vaswani, “Attention is All You Need,” in *Conference on Neural Information Processing Systems (NIPS)* (ACM, 2017).
8. L. Köhs, B. Alt, and H. Koepl, “Markov Chain Monte Carlo for Continuous-Time Switching Dynamical Systems,” in *ICML* (International Machine Learning Society, 2022).
9. J. R. Norris, *Markov Chains*, ser. Cambridge Series in Statistical and Probabilistic Mathematics (Cambridge University Press, 1997).
10. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016).
11. C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, “Hierarchical convolutional features for visual tracking,” in *International Conference on Computer Vision (ICCV)* (IEEE, 2015).
12. S. Särkkä and A. Solin, *Applied Stochastic Differential Equations*, ser. Institute of Mathematical Statistics Textbooks (Cambridge University Press, 2019).
13. J. L. Doob, “Markoff Chains—Denumerable Case,” *Transactions of the American Mathematical Society* 58, no. 3 (1945): 455–473.
14. Y. Wu, J. Lim, and M.-H. Yang, “Online Object Tracking: A Benchmark,” in *Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2013).
15. H. Fan, L. Lin, F. Yang, et al., “Lasot: A High-Quality Benchmark for Large-Scale Single Object Tracking,” arXiv abs/1809.07845 (2018).
16. M. Mueller, N. Smith, and B. Ghanem, “A Benchmark and Simulator for UAV Tracking,” in *European Conference on Computer Vision (ECCV)* (Springer, 2016).
17. M. Kristan, J. Matas, A. Leonardis, et al., “A Novel Performance Evaluation Methodology for Single-Target Trackers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, no. 11 (Nov 2016): 2137–2155.
18. H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, “Need for Speed: A Benchmark for Higher Frame Rate Object Tracking,” in *International Conference on Computer Vision (ICCV)* (IEEE, 2017).
19. B. Yan, H. Peg, J. Fu, D. Wang, and H. Lu, “Learning Spatio-Temporal Transformer for Visual Tracking,” in *International Conference on Computer Vision (ICCV)* (IEEE, 2021).
20. X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, “Transformer Tracking,” in *Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2021).
21. Z. Fu, Q. Liu, Z. Fu, and Y. Wang, “Stmtrack: Template-Free Visual Tracking with Space-Time Memory Networks,” in *Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2021).
22. Z. Zhou, W. Pei, X. Li, H. Wang, F. Zheng, and Z. He, “Saot: Saliency-Associated Object Tracking,” in *International Conference on Computer Vision (ICCV)* (IEEE, 2021).
23. Z. Pi, W. Wan, C. Sun, C. Gao, N. Sang, and C. Li, “Hierarchical Feature Embedding for Visual Tracking,” in *European Conference on Computer Vision (ECCV)*, ed. S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner (Springer, 2022).
24. C. Mayer, M. Danelljan, G. Bhat, et al., “Transforming Model Prediction for Tracking,” in *Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2022).
25. X. Li, Y. Huang, Z. He, Y. Wang, H. Lu, and M.-H. Yang, “Cite-tracker: Correlating Image and Text for Visual Tracking,” in *International Conference on Computer Vision (ICCV)* (IEEE, 2023).
26. F. Xie, Z. Wang, and C. Ma, “Diffusiontrack: Point Set Diffusion Model for Visual Object Tracking,” in *Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2024).
27. L. Hong, S. Yan, R. Zhang, et al., “Onetracker: Unifying Visual Object Tracking with Foundation Models and Efficient Tuning,” in *Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2024).
28. N. Wang, W. Zhou, J. Wang, and H. Li, “Transformer Meets Tracker: Exploiting Temporal Context for Robust Visual Tracking,” in *Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2021).
29. B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, “SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks,” in *Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2018).
30. G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, “Learning Discriminative Model Prediction for Tracking,” in *International Conference on Computer Vision (ICCV)* (IEEE, 2019).
31. H. Fan and H. Ling, “Cract: Cascaded Regression Align-Classification for Robust Visual Tracking,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2021).
32. G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, “Unveiling the Power of Deep Tracking,” in *European Conference on Computer Vision (ECCV)* (Springer, 2018).
33. F. Xie, N. Wang, Y. Yao, W. Yang, K. Zhang, and B. Liu, “Hierarchical Representations with Discriminative Meta-Filters in Dual Path Network for Tracking,” in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, ed. Y. Peng, Q. Liu, H. Lu, Z. Sun, C. Liu, X. Chen, H. Zha, and J. Yang (IEEE, 2020).
34. H. Saribas, H. Cevikalp, O. Köpüklü, and B. Uzun, “TRAT: Tracking by Attention Using Spatio-Temporal Features,” CoRR abs/2011.09524 (2020).