ARTrackV2: Prompting Autoregressive Tracker Where to Look and How to Describe

Yifan Bai Zeyang Zhao Yihong Gong Xing Wei ✉

Xi'an Jiaotong University

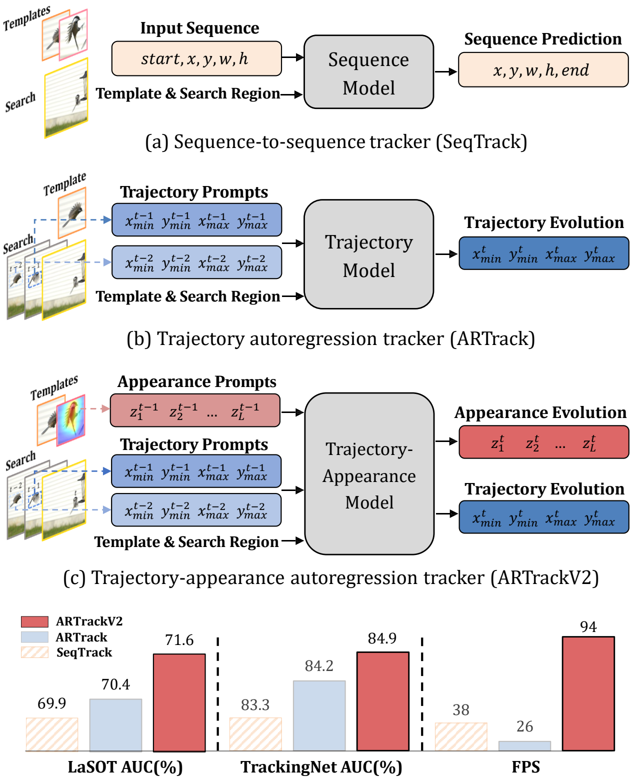
{yfbai, zeyang}@stu.xjtu.edu.cn {ygong, weixing}@mail.xjtu.edu.cn

https://ARTrackV2.github.io/

Abstract

We present ARTrackV2, which integrates two pivotal aspects of tracking: determining ***where to look*** *(localization) and* ***how to describe*** *(appearance analysis) the target object across video frames. Building on the foundation of its predecessor, ARTrackV2 extends the concept by introducing a unified generative framework to* ***"read out"*** *object's trajectory and* ***"retell"*** *its appearance in an autoregressive manner. This approach fosters a time-continuous methodology that models the joint evolution of motion and visual features, guided by previous estimates. Furthermore, ARTrackV2 stands out for its efficiency and simplicity, obviating the less efficient intra-frame autoregression and hand-tuned parameters for appearance updates. Despite its simplicity, ARTrackV2 achieves state-of-the-art performance on prevailing benchmark datasets while demonstrating remarkable efficiency improvement. In particular, ARTrackV2 achieves AO score of 79.5% on GOT-10k, and AUC of 86.1% on TrackingNet while being* 3.6× *faster than ARTrack. The code will be released.*

1. Introduction


Templates
Search
Input Sequence
$start, x, y, w, h$
Template & Search Region
Sequence Model
Sequence Prediction
x, y, w, h, end
Template
Search
Trajectory Prompts
x^{t-1}_{min} y^{t-1}_{min} x^{t-1}_{max} y^{t-1}_{max}
x^{t-2}_{min} y^{t-2}_{min} x^{t-2}_{max} y^{t-2}_{max}
Template & Search Region
Trajectory Model
Trajectory Evolution
x^{t}_{min} y^{t}_{min} x^{t}_{max} y^{t}_{max}
Templates
Search
Appearance Prompts
z^{t-1}_{1} z^{t-1}_{2} ... z^{t-1}_{L}
Trajectory Prompts
x^{t-1}_{min} y^{t-1}_{min} x^{t-1}_{max} y^{t-1}_{max}
x^{t-2}_{min} y^{t-2}_{min} x^{t-2}_{max} y^{t-2}_{max}
Template & Search Region
Trajectory-Appearance Model
Appearance Evolution
z^{t}_{1} z^{t}_{2} ... z^{t}_{L}
Trajectory Evolution
x^{t}_{min} y^{t}_{min} x^{t}_{max} y^{t}_{max}
ARTrackV2
ARTrack
SeqTrack
69.9
70.4
71.6
83.3
84.2
84.9
38
26
94
LaSOT AUC(%)
TrackingNet AUC(%)
FPS

(a) Sequence-to-sequence tracker (SeqTrack)

(b) Trajectory autoregression tracker (ARTrack)

(c) Trajectory-appearance autoregression tracker (ARTrackV2)

(d) Performance comparison

Figure 1. **Frameworks and performance comparison** of trackers following the sequence generation paradigm. (a) SeqTrack views tracking as sequence prediction. (b) ARTrack introduces trajectory evolution. (c) ARTrackV2 incorporates joint trajectory-appearance evolution. (d) Performance comparison.

Visual object tracking [6, 23, 33, 35, 39, 48], a cornerstone in the realm of computer vision, has witnessed transformative advancements over the past decade. Its applications span a diverse array of fields, from autonomous vehicles to surveillance, and from augmented reality to human-computer interaction. At its core, visual tracking involves the continuous localization of an object within a video sequence, typically initiated from the first frame.

In this research area, previous approaches have primarily focused on either trajectory estimation or appearance modeling. Traditional methods, such as the application of Kalman filters [4, 8, 62] and Particle filters [2, 29], emphasize predicting the object's motion by leveraging historical states. In contrast, modern learning-based methods strive to understand and track the visual features of the target object, often employing a template-matching framework. However, these approaches typically adopt frame-level training strategies, overlooking the temporal dependencies across frames. Some methods attempt to handle appearance changes over time using dynamic templates, which are updated using heuristic rules [24] or learnable modules [13, 14, 63].

The recent shift towards a generative paradigm [9] in visual tracking [11, 60], conceptualizing the task as sequence generation, has set new performance benchmarks. This approach simplifies the process, directly predicting object coordinates sequentially. SeqTrack [11], as shown in Figure 1(a), introduces an *intra-frame* sequence model that generates four tokens of the bounding box autoregressively. It also showcases that prepending previous coordinate tokens in inference could improve accuracy further. In contrast, ARTrack [60] concentrates on *inter-frame* autoregression (Figure 1(b)). It advocates for video sequence-level training (rather than frame-level) to maintain consistency between training and testing in terms of data distributions and task objectives [36]. Overall, this generative framework has its flexibility to utilize historical trajectory tokens, referred to as *trajectory prompts*, to model trajectory evolution continually.

In this paper, we go one step further and introduce a *joint trajectory-appearance* autoregression tracker. Building upon the foundation of its predecessor, ARTrackV2 extends the concept by implementing a unified generative framework that models the evolution of both trajectory and appearance. The intuition behind this idea is simple: **if the tracker successfully tracks an object, it should not only "read out" object's position but also "retell" its appearance**. Alongside the time-series modeling of trajectory proposed by ARTrack, we maintain an autoregressive model to simultaneously reconstruct the object's appearance, using a set of *appearance prompts*, as illustrated in Figure 1(c). These tokens, on the one hand, function similarly to dynamic templates, interacting with the search region through attention mechanisms. Beyond that, they are trained to rebuild the object's appearance, requiring an understanding of visual feature evolution. We design a masking strategy that intentionally prevents the attention from appearance tokens to trajectory ones, preventing the appearance model from merely cropping visual features based on the predicted trajectory.

Furthermore, ARTrackV2 distinguishes itself through its operational simplicity and efficiency. Different from SeqTrack and ARTrack, it utilizes a pure encoder architecture to process all tokens within a frame, in parallel. ARTrackV2 abandons intra-frame autoregression that impedes tracking efficiency while maintaining the time-autoregressive framework (aka, inter-frame autoregression). In contrast to many contemporary tracking systems that require multiple training stages [13, 14, 63] or hand-tuned parameters for template updates [11, 24], ARTrackV2 undergoes end-to-end training within a single stage. This approach yields outstanding performance on various benchmark datasets, with the base model achieving an impressive AUC score of 71.6% on LaSOT and 84.9% on TrackingNet. Notably, it accomplishes this while exhibiting a substantial 3.6× speed improvement compared to ARTrack, as demonstrated in Figure 1(d). our top-performing model achieves an even higher AUC score of 73.6% on LaSOT and an impressive 86.1% on TrackingNet, significantly outperforming SeqTrack and ARTrack while maintaining remarkable speed improvements of approximately 3× to 5×.

To summarize, ARTrackV2 enhances its predecessor in the following ways:

- **Extend the concept**: we complement the generative framework for visual tracking to encompass both trajectory generation and appearance reconstruction.
- **Strengthen inter-frame autoregression**: we uphold the time-autoregressive model to jointly evolve trajectory and appearance. Also, we introduce sequence data augmentation to improve accuracy.
- **Eliminate intra-frame autoregression**: we employ a pure encoder architecture that enables parallel processing of all tokens within a frame, moving away from the less efficient intra-frame autoregressive decoder.

2. Related Work

Tracking Framework. Prevailing trackers [3, 12, 15, 16, 27, 37, 73] often employing a template-matching framework reference template to match target within the search region. Initially, these approaches employ a backbone to integrate visual features [7, 13, 41, 66], then split tracking into multiple subtasks [1, 5, 38, 50, 52, 72] such as object scale estimation and center point localization, divide and conquer with specific heads. Meanwhile, they introduce complex post-processings, overlooking potential temporal dependencies. Recently, generative paradigm [11, 60] redefines tracking as a sequence generation task. After visual integration, this approach unified multiple tracking subtasks as an intra-frame sequence model in an autoregressive manner. This methodology simplified the tracking framework and leveraged preceding trajectory tokens to model trajectory evolution but impeded efficiency by introducing intra-frame autoregression. Thus, we propose a pure encoder architecture that enables parallel processing of all tokens, abandoning intra-frame autoregression while preserving a time-autoregression nature.

Appearance Modeling. To handle the appearance variation that often occurs in tracking scenarios, typical discriminative approaches [13, 14, 63] leverage a score model trained to discriminate whether the tracked region contains the target. Recently, SeqTrack [11] introduced a likelihood-based strategy that uses the likelihood of generated tokens to select dynamic templates without incremental training. Moreover, the above methods rely on hand-tuned parameters such as update interval and threshold for specific benchmarks [3, 15, 73]. Furthermore, both of them model appearance in discrete frames [5, 16, 40, 56, 64, 65, 69]. In contrast, we present generative autoregressive appearance reconstruction to model appearance evolution in successive video with end-to-end single-stage training, fully exploiting the temporal potential.

3. Method

3.1. Revisiting ARTrack

ARTrack [60] constitutes a sequence generation framework for visual tracking, with its primary focus centered on the generation of time-series coordinates. This is achieved by utilizing a shared vocabulary to tokenize the object's trajectory, representing it as a discrete sequence of coordinates. Subsequently, the framework employs an encoder-decoder architecture to assimilate visual information and progressively model the sequential evolution of the trajectory prompted by preceding coordinate tokens. This modeling is expressed as a conditional probability:

$$P\left(\boldsymbol{Y}^t|\boldsymbol{Y}^{t-N:t-1},(\boldsymbol{C},\boldsymbol{Z},\boldsymbol{X}^t)\right), \tag{1}$$

Here, $\boldsymbol{Z}$ and $\boldsymbol{X}^t$ represent the given template and search images at time step t, $\boldsymbol{C}$ serves as the command token, and $\boldsymbol{Y}$ signifies the target sequence associated with $\boldsymbol{X}$.

Beyond frame-level training and optimization, ARTrack is learned over video sequence with structure objectives to obviate bias between the training and testing phases in data distributions and task objectives. Furthermore, a task-specific SIoU loss [26] is utilized to enhance accuracy.

Motivation. At the core of tracking lies the challenge of determining where to focus attention and how to describe the target accurately. ARTrack offers valuable insights by emphasizing the importance of continuous trajectory evolution, allowing for precise "reading out" of the object's position. However, it falls short in effectively "retelling" the object's changing appearance over time. Furthermore, ARTrack employs an approach known as intra-frame autoregression, which involves generating four tokens of the bounding box sequentially. This intra-frame autoregression method significantly hampers tracking efficiency. To address these limitations, we introduce ARTrackV2, which leverages joint trajectory-appearance evolution and utilizes a pure encoder architecture to enhance processing speed.

3.2. Joint Trajectory-Appearance Autoregression

The framework of ARTrackV2 is depicted in Figure 2. We expand upon the concept introduced in ARTrack by synchronously modeling the evolution of both trajectory and appearance, thus reinforcing inter-frame autoregression. This is formulated as a probability expression:

$$P\left(\boldsymbol{Y}^t,\boldsymbol{Z}^t|\boldsymbol{Y}^{t-N:t-1},\boldsymbol{Z}^{t-1},(\boldsymbol{C},\boldsymbol{Z}^0,\boldsymbol{X}^t)\right), \tag{2}$$

Here, $\boldsymbol{Z}^0$ represents the initial template, which remains static throughout tracking. The appearance tokens function as dynamic templates, denoted as $\boldsymbol{Z}^{t-1}$ and $\boldsymbol{Z}^t$ in red, effectively describing the temporal evolution of the target's appearance in a continuous manner.

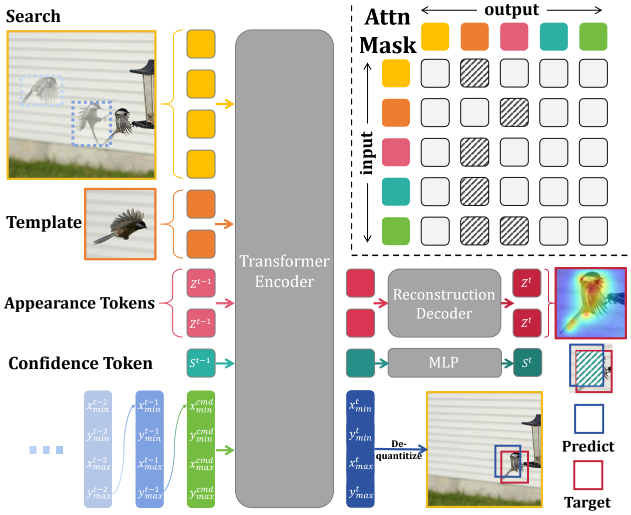


Figure 2. **ARTrackV2 framework**. Initially, we utilize a Transformer encoder to process all tokens within a frame in parallel, with a masking strategy shown on the top right. Subsequently, appearance tokens are directed to a reconstruction decoder, where the object's appearance within the ongoing search region is reconstructed. Simultaneously, the confidence token is fed into an MLP to predict the IoU between the estimated and ground truth bounding boxes, serving as a measure of the quality of appearance tokens.

Pure Encoder Architecture. In the context of generative paradigm trackers [41, 66], there is an efficiency drawback associated with intra-frame autoregression when compared to prevailing tracking methods. Therefore, in our pursuit of simplicity and efficiency, we opt for a transformer encoder [20, 30] that can process all tokens within a frame in parallel. Initially, both the template and search images are divided into patches, flattened, and projected to form a sequence of token embeddings. Similar to ARTrack, we map the object's trajectory across frames to a common coordinate system and tokenize it as trajectory prompts, utilizing a shared vocabulary [9, 10, 61]. We then concatenate visual tokens, trajectory tokens, and four command tokens (each representing one of the four bounding box tokens), add positional and identity embeddings, and input them into the transformer encoder. This approach eliminates the need for intra-frame autoregression while still maintaining the time-autoregressive nature of the framework, thereby improving overall efficiency.

Autoregressive Appearance Reconstruction. We employ a set of appearance tokens alongside a reconstructed decoder to recreate the target's appearance within the current search region. These appearance tokens, termed "appearance prompts", operate akin to dynamic templates. For each video clip, they initialize as the template within the first frame. In each subsequent frame, they interact with the current search region to extract the target's appearance, through the transformer encoder. Then appearance tokens enter the reconstructed decoder, which rebuilds the target's appearance formed as the feature map of the search region cropped based on the object's position. The output from the reconstructed decoder continuously updates the appearance tokens, which propagates into subsequent frames. However, a challenge arises when the target becomes invisible, either due to being out of view or significantly occluded. In such instances, invisible appearance propagation can erroneously guide the model to "read out" nonsensical target localization in the following frames. To address this, we instruct the appearance tokens to maintain their current state in scenarios devoid of visual cues, to prevent unwarranted appearance evolution, ensuring accurate model behavior. This process allows the model to capture appearance variation over time while preserving its autoregressive nature.

Appearance Evolution Indicator. In scenarios involving complex conditions such as full occlusion, improper evolution of appearance can result in the loss of target. To tackle this challenge, we propose a solution that guides the model's evolution of appearance with an indicator. Our approach employs a learnable confidence token and a confidence prediction module comprises a three-layer perceptron. Moreover, we adopt Intersection over Union (IoU) as the indicator's metric based on the fact that it aligns with common tracking evaluation metrics. In continuous frames, the confidence token interacts with all tokens through the transformer encoder. This interaction implicitly guides the appearance tokens regarding whether to evolve or maintain their current state. Subsequently, the confidence token feeds into the perceptron, predicting the IoU between the model's estimations and the ground truth boxes. Like appearance tokens, the estimated indicator updates the previous confidence token and propagates into the subsequent frame.

Oriented Masking. To prevent the model from exclusively fixating on cropping visual features solely based on predicted localization and overlooking the understanding of appearance evolution, we implement an attention masking strategy within the transformer encoder. Beyond MixFormer [13], which concerns intrinsic characteristics within templates to eliminate potential interactive distractors, our approach involves restricting appearance tokens. We force appearance tokens to solely interact with the search region (for reconstructing the target's appearance) and confidence token (for instructing appearance evolution). This deliberate process aims to deter appearance tokens from merely cropping visual features based on target localization and then limits the comprehension of appearance evolution.

3.3. Sequence Augmentation

When compared to frame-level training, which involves sampling image pairs from videos [3, 37, 55], sequence-level training [36, 60] aligns the training and testing data distributions by exclusively sampling successive video clips instead of individual image pairs. However, this approach results in a sharp reduction in the amount of training data available. To overcome this challenge, we investigate sequence-level augmentation methods.

Drawing inspiration from multiple object tracking techniques [49, 68, 71], we experiment with fixed and random interval sampling, but both of these methods negatively impact tracking accuracy. We observe that these approaches disrupt the natural progression of temporal information, leading the tracker to learn spurious temporal interactions.

As a result, our criterion for designing augmentation is to preserve the time-series nature of the data. We propose a straightforward yet effective augmentation strategy known as "reverse augmentation". Given a video sequence, we invert it with a certain probability to expand the training dataset.

3.4. Training and Inference

ARTrackV2 emphasizes video sequence-level training and facilitates joint trajectory-appearance evolution in an end-to-end manner.

Training. Similar to its predecessor, ARTrackV2 undergoes sequence-level training. We employ a structured objective that maximizes the log-likelihood of trajectory sequences. Moreover, we incorporate a task-agnostic SIoU loss [26] to enhance the measurement of spatial correlation.

While modeling the trajectory over time, ARTrackV2 also maintains an autoregressive model to reconstruct appearance. Drawing inspiration from MAE [30], we introduce reconstruct tokens masking strategy, after the transformer encoder processing, we sample a subset of appearance tokens and mask them. This creates a challenging task aimed at preventing the overfitting of reconstruction. Subsequently, we compute the mean squared error (MSE) between the reconstructed tokens and the target within the search region or the preceding appearance tokens, depending on whether the object is visible. To avoid poor-quality appearance evolution, we introduce the confidence prediction module, trained by L1 loss between the actual and predicted IoU.

For each video clip, the cached trajectory prompts are initialized with the bounding box from the first frame, and the appearance tokens are set to match the template. Both the trajectory-appearance prompts and the confidence token are iteratively propagated into subsequent frames in an autoregressive manner. The overall tracker is optimized by sequence-level loss function, which is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda_{\text{SIoU}}\mathcal{L}_{\text{SIoU}} + \lambda_{\text{mse}}\mathcal{L}_{\text{mse}} + \lambda_{\text{L1}}\mathcal{L}_{\text{L1}}, \tag{3}$$

where $\mathcal{L}_{\text{ce}}$, $\mathcal{L}_{\text{SIoU}}$, $\mathcal{L}_{\text{mse}}$ and $\mathcal{L}_{\text{L1}}$ are the cross-entropy loss, SIoU loss, MSE loss, and IoU L1 loss respectively. The λ values serve as weights to balance the contribution of each loss term.

Inference. During inference, we initialize the trajectory and appearance tokens as previously described. Subsequently, we simultaneously generate the trajectory sequence from the estimated likelihood with `argmax` sampling and reconstruct the target's appearance. This process is carried out in an autoregressive manner, where the trajectory, appearance, and confidence token are iteratively propagated into subsequent frames. It's worth noting that, during the reconstruction process, we intentionally refrain from appearance tokens masking, unlike the training phase.

4. Experiments

4.1. Implementation Details

The models train with 8 NVIDIA RTX A6000 GPUs, with training times ranging from approximately 26 to 120 hours, depending on the specific experimental configurations.

Model Variants. We trained three variants of ARTrackV2 with different configurations as follows:

- **ARTrackV2$_{256}$.** Backbone: ViT-Base; Template size: [128×128]; Search region size: [256×256];
- **ARTrackV2$_{384}$.** Backbone: ViT-Base; Template size: [192×192]; Search region size: [384×384];
- **ARTrackV2-L$_{384}$.** Backbone: ViT-Large; Template size: [192×192]; Search region size: [384×384].

Training Strategy. We adhere to established protocols for training and evaluating our models, consistent with ARTrack. The training dataset includes GOT-10k [32] (with 1k sequences removed from the GOT-10k train split, as per [63]), TrackingNet [47], and LaSOT [22]. To ensure a fair evaluation of the GOT-10k test set, our models learn from the entire GOT-10k training split following its one-shot protocol.

The models are optimized using AdamW [44] with a weight decay of 5×10^{-2}. The learning rate for the backbone is set to 8×10^{-6}, while other parameters use a learning rate of 8×10^{-5}. The training process comprises 60 epochs, with 960 video sequences in each epoch. Each sequence consists of 32 frames, constrained by GPU memory limitations.

Furthermore, in line with ARTrack [60], and to align with established trackers [11, 25, 31, 51] that are trained using image datasets such as COCO2017 [42], we employ frame-level training to pre-train our models. During this process, we utilize four training datasets and apply image data augmentation techniques, including horizontal flip and brightness jittering, which are consistent with OSTrack [66] and SeqTrack [11]. The pre-trained models are optimized using AdamW with a weight decay of 10^{-4}, with the learning rate for the backbone and other parameters set the same as previously mentioned. Our pre-trained model undergoes 240 epochs of training, with 60k matching pairs processed per epoch.

4.2. Main Results

We evaluate the performance of our proposed ARTrackV2$_{256}$ and ARTrackV2-L$_{384}$ on several benchmarks, including GOT-10k [32], TrackingNet [47], LaSOT [22] and LaSOT$_{\text{ext}}$ [21].

GOT-10k [32]. GOT-10k is a comprehensive generic object tracking dataset comprising video sequences featuring real-world moving objects with manually annotated bounding boxes. The dataset advocates for a one-shot protocol, which necessitates that trackers are trained exclusively on the GOT-10k training split to ensure that the object classes in the training and testing sets do not overlap. Adhering to this protocol, our ARTrackV2 is trained exclusively on the GOT-10k training split and evaluated on the test set. As demonstrated in Table 1, our ARTrackV2-L$_{384}$ outperforms state-of-the-art trackers across all metrics. Notably, our ARTrackV2$_{256}$ and ARTrackV2$_{384}$ surpasses other trackers with higher resolution and larger backbones except ARTrack.

TrackingNet [47]. TrackingNet is an extensive tracking dataset comprising over 30,000 videos that cover a wide range of real-world scenarios and content. Each video is annotated with manually labeled bounding boxes. We assess the performance of ARTrackV2 on its test set which contains 511 videos covering diverse object categories, as illustrated in Table 1. This table shows that not only does our ARTrackV2$_{384}$ outperform all other trackers in AUC, but our ARTrackV2-L$_{384}$ also establishes a new state-of-the-art in three matrices on this large-scale benchmark.

LaSOT [22]. LaSOT is a benchmark designed for long-term tracking, comprising 280 videos in its test set, effectively assessing the tracker's robustness in extended video sequences. Table 1 demonstrates that our ARTrackV2$_{256}$ achieves comparable performance to ARTrack$_{384}$, despite having lower input resolution. Furthermore, our ARTrackV2-L$_{384}$ significantly enhances performance, setting a new state-of-the-art while running at 49 FPS, which is over 5× faster than SeqTrack-L$_{384}$ (9 FPS).

LaSOT$_{\text{ext}}$ [21]. LaSOT$_{\text{ext}}$ serves as an extension of LaSOT, including an additional 150 videos. These new sequences introduce challenging tracking scenarios, such as occlusions and variations in small objects. To demonstrate the robustness of our models in handling these difficult scenarios, we evaluate ARTrackV2 and present the results in Table 1. Remarkably, our ARTrackV2$_{384}$ achieves exceptional accuracy,

Methods	GOT-10k*			TrackingNet			LaSOT			LaSOText		
	AO(%)	$SR_{0.5}$(%)	$SR_{0.75}$(%)	AUC(%)	P_{Norm}(%)	P(%)	AUC(%)	P_{Norm}(%)	P(%)	AUC(%)	P_{Norm}(%)	P(%)
SiamFC$_{255}$ [3]	34.8	35.3	9.8	57.1	66.3	53.3	33.6	42.0	33.9	23.0	31.1	26.9
ECO$_{224}$ [17]	31.6	30.9	11.1	55.4	61.8	49.2	32.4	33.8	30.1	22.0	25.2	24.0
DiMP$_{288}$ [45]	61.1	71.7	49.2	74.0	80.1	68.7	56.9	65.0	56.7	39.2	47.6	45.1
SiamR-CNN$_{255}$ [55]	64.9	72.8	59.7	81.2	85.4	80.0	64.8	72.2	-	-	-	-
Ocean$_{255}$ [72]	61.1	72.1	47.3	-	-	-	56.0	65.1	56.6	-	-	-
TrDiMP$_{352}$ [58]	67.1	77.7	58.3	78.4	83.3	73.1	63.9	-	61.4	-	-	-
SLT-TrDiMP$_{352}$ [36]	67.5	78.8	58.7	78.1	83.1	73.1	64.4	73.5	-	-	-	-
TransT$_{256}$ [12]	67.1	76.8	60.9	81.4	86.7	80.3	64.9	73.8	69.0	-	-	-
STARK$_{320}$ [63]	68.8	78.1	64.1	82.0	86.9	-	67.1	77.0	-	-	-	-
SwinTrack-B$_{384}$ [41]	72.4	80.5	67.8	84.0	-	82.8	71.3	-	76.5	49.1	-	55.6
MixFormer-L$_{320}$ [13]	-	-	-	83.9	88.9	83.1	70.1	79.9	76.3	-	-	-
OSTrack$_{384}$ [66]	73.7	83.2	70.8	83.9	88.5	83.2	71.1	81.1	77.6	50.5	61.3	57.6
CTTrack-B$_{320}$ [51]	71.3	80.7	70.3	82.5	87.1	80.3	67.8	77.8	74.0	-	-	-
CTTrack-L$_{320}$ [51]	72.8	81.3	71.5	84.9	89.1	83.5	69.8	79.7	76.2	-	-	-
TATrack-B$_{224}$ [31]	73.0	83.3	68.5	83.5	88.3	81.8	69.4	78.2	74.1	-	-	-
TATrack-L$_{384}$ [31]	-	-	-	85.0	89.3	84.5	71.1	79.1	76.1	-	-	-
GRM-B$_{256}$ [25]	73.4	82.9	70.4	84.0	88.7	83.3	69.9	79.3	75.8	-	-	-
GRM-L$_{320}$ [25]	-	-	-	84.4	88.9	84.0	71.4	81.2	77.9	-	-	-
MixViT$_{320}$ [13]	72.5	82.4	69.9	83.5	88.3	82.0	69.6	79.9	75.9	-	-	-
MixViT-L$_{320}$ [13]	75.7	85.3	75.1	85.4	90.2	85.7	72.4	82.2	80.1	-	-	-
SeqTrack-B$_{256}$ [11]	74.7	84.7	71.8	83.3	88.3	82.2	69.9	79.7	76.3	49.5	60.8	56.3
SeqTrack-L$_{384}$ [11]	74.8	81.9	72.2	85.5	89.8	85.8	72.5	81.5	79.3	50.7	61.6	57.5
ARTrack$_{256}$ [60]	73.5	82.2	70.9	84.2	88.7	83.5	70.4	79.5	76.6	46.4	56.5	52.3
ARTrack$_{384}$ [60]	75.5	84.3	74.3	85.1	89.1	84.8	72.6	81.7	79.1	51.9	62.0	58.5
ARTrack-L$_{384}$ [60]	78.5	87.4	77.8	85.6	89.6	86.0	73.1	82.2	80.3	52.8	62.9	59.7
ARTrackV2$_{256}$	75.9	85.4	72.7	84.9	89.3	84.5	71.6	80.2	77.2	50.8	61.9	57.7
ARTrackV2$_{384}$	77.5	86.0	75.5	85.7	89.8	85.5	73.0	82.0	79.6	52.9	63.4	59.1
ARTrackV2-L$_{384}$	**79.5**	**87.8**	**79.6**	**86.1**	**90.4**	**86.2**	**73.6**	**82.8**	**81.1**	**53.4**	**63.7**	**60.2**

Table 1. State-of-the-art comparison on GOT-10k [32], TrackingNet [47], LaSOT [22] and LaSOText [21]. Where * denotes only trained on GOT-10k. The number in the subscript denotes the search region resolution. Best in **bold**, second best underlined, and third best underwave.

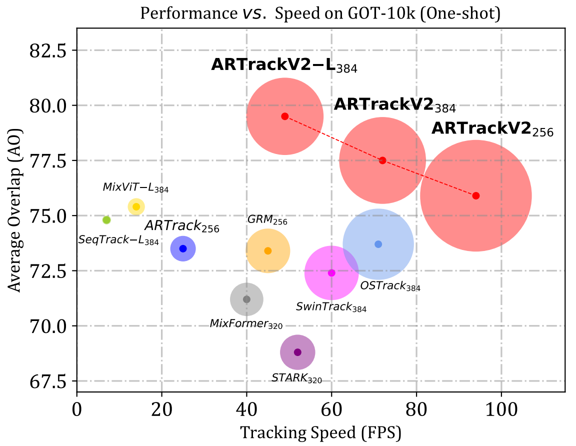


Figure 3. Comparison of accuracy vs. latency trade-off for different tracking methods in GOT-10k (one-shot setting).

surpassing other trackers with larger backbones with 72 FPS. Furthermore, our ARTrackV2-L$_{384}$ outperforms ARTrack-L$_{384}$ and establishes a new state-of-the-art performance, operating at a speed of 49 FPS, which is 3× faster than ARTrack.

4.3. Accuracy vs. Latency

In comparison to ARTrack, we have eliminated intra-frame autoregression to enhance inference speed, resulting in nearly a 3× improvement in inference efficiency without compromising accuracy. To illustrate this improvement, we conducted a comparative analysis of state-of-the-art trackers on GOT-10k using the one-shot protocol, as depicted in Figure 3. Our ARTrackV2-L$_{384}$ achieves a new state-of-the-art with an impressive AO of 79.5%, while ARTrackV2$_{256}$ delivers competitive performance at 94 FPS, surpassing other trackers with higher resolutions and larger backbones.

4.4. Experimental Analyses

We analyze the main properties of the ARTrackV2. For the following experimental studies, we follow the GOT-10k test protocol unless otherwise noted. Default settings are marked in gray .

Summary of Cumulative Effect. We conduct comprehensive ablation studies to analyze our proposed approach, considering several key aspects: evaluating the impact of the pure encoder architecture, assessing the effectiveness of

model variants	AO	$SR_{0.5}$	$SR_{0.75}$	FPS
ARTrack [60]	73.5	82.2	70.9	26
pure encoder architecture	71.0	79.9	68.2	116
+ appearance evolution	74.2	83.1	71.4	98
+ confidence prediction	74.7	83.7	72.1	94
+ masking strategy	75.2	84.8	72.4	94
+ sequence augmentation	**75.9**	**85.4**	**72.7**	**94**

Table 2. Summary of **cumulative effects**.

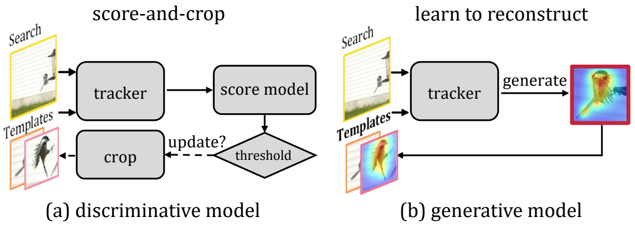
autoregressive appearance evolution, examining the contribution of the confidence prediction module, validating the masking strategy, and exploring the benefits of sequence-level data augmentation. Additionally, we systematically evaluate the cumulative effects of integrating these various components, and the results are summarized in Table 2.

We observed that adopting the pure encoder architecture significantly improved tracking efficiency. However, this improvement came at the cost of a decrease in accuracy, which we attributed to the lack of intra-frame temporal information. To address this, we introduced autoregressive appearance evolution which leveraged appearance reconstructions, confidence prediction, attention masking strategy, and sequence data augmentation to strengthen inter-frame autoregression. These modifications teach the model to "retell" the object's appearance variation in a time-autoregressive manner. As a result of these enhancements, we complement generative paradigm trackers to joint evolution of trajectory and appearance, achieving substantial improvements in model performance, and ultimately establishing state-of-the-art results.

appearance model	single-stage	thresholds tuning	AO
discriminative (score-based)	✗	✓	74.5
discriminative (likelihood-based)	✓	✓	74.1
generative (reconstruction)	✓	✗	**75.9**

Table 3. **Appearance model comparison**.

Appearance Model. In this subsection, we delve into the generative appearance model in ARTrackV2. This model differs from previous discriminative models, which determine whether the cropped region from the search image, using the tracking result, is reliable for updating the template. Discriminative approaches typically require an additional stage to train a score model [13, 14, 63] to classify whether the tracked region contains the target object, thus breaking the single-stage end-to-end learning framework. SeqTrack introduces a likelihood-based strategy that uses the likelihood of the generated coordinate tokens to select dynamic templates in a single stage. Moreover, these methods often involve hand-tuning parameters for each individual dataset, including score/likelihood thresholds and update frequency. As depicted in Figure 4, ARTrackV2 employs a simple unified approach to appearance evolution. Instead of score-and-crop, it learns to recreate the template in a continuous autoregressive manner. It also adopts a masking strategy to prevent attention from appearance tokens to the trajectory ones. We compare these different appearance models in Table 3, demonstrating that our generative model performs better than score-based or likelihood-based discriminative approaches.



(a) discriminative model

(b) generative model

Figure 4. **Comparison of appearance modeling approaches**. (a) discriminative model adopts a score-and-crop strategy to decide updates. (b) generative model learns to reconstruct the template.

reconstruction objective	AO	$SR_{0.5}$	$SR_{0.75}$
image reconstruction	74.8	83.9	71.1
feature reconstruction	**75.9**	**85.4**	**72.7**

Table 4. **Reconstruction objective**.

Reconstruction Objective. The quality of the appearance tokens is ascertained through an adequate reconstruction objective of the target, which can be in the image pixel domain or in the latent feature domain. To investigate this, we conducted exploratory experiments, as outlined in Table 4. Our findings indicate that "feature reconstruction" leads to a noteworthy improvement of approximately 1.1% on the AO metric. This improvement demonstrates that feature reconstruction is more effective in appearance evolution. In contrast, image reconstruction may tend to excessively focus on intricate details or background information. In scenarios characterized by motion blur and occlusion settings, this approach encounters challenges in accurately reconstructing the target at the pixel level.

indicator metric	AO	$SR_{0.5}$	$SR_{0.75}$
w/o	75.1	84.7	71.9
confidence	74.9	84.4	71.5
distance	75.1	84.5	72.2
visibility	74.6	84.2	71.3
IoU	**75.9**	**85.4**	**72.7**

Table 5. **Appearance evolution indicator**.

Appearance Evolution Indicator. To ensure quality appearance evolution, it is necessary to employ indicators that

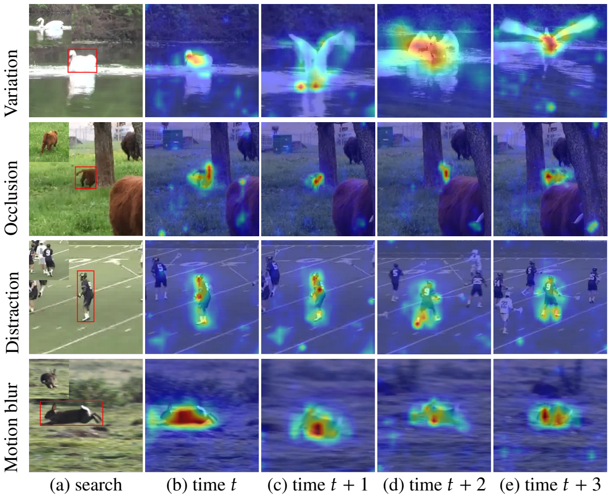


Figure 5. **Attention visualization**. (a): Search region and template. The red boxes denote the ground truth. (b)-(e): Appearance tokens to search the cross-attention map of ARTrackV2.

guide the reconstruction of appearance tokens. These indicators serve to characterize the evolution quality. In our analysis of different metrics' impact on the model, we present the findings in Table 5. The metric labeled "confidence" [13,63] denotes the confidence assigned to the current template, indicating whether it contains the target. The "distance" [24] represents the cosine distance between the features of the appearance tokens and the target's appearance feature within the ongoing search region. The "visibility" [32] quantifies the visibility ratio of the target in the search region. Lastly, the "IoU" metric measures the Intersection over Union between the predicted and the ground truth bounding boxes.

Contrary to previous research findings [13, 15, 24, 28, 63, 70], our investigation has revealed that employing IoU as the reconstruction indicator leads to superior accuracy. This observation can be primarily attributed to the fact that when compared to alternative methods, the IoU metric is more closely aligned with the evaluation metric utilized for tracking. Consequently, it provides a precise reflection of the quality of evolution. We also note that the visibility metric yields unsatisfactory results. GOT-10k provides an assessment of object visibility segmented into 9 levels, but the boundaries between these levels are often vague and may contain noisy labels. This poses a challenge for models in precisely evaluating the target's visibility.

Visualization and Analysis. To gain deeper insights into the autoregressive appearance evolution, we generate cross-attention maps about the appearance tokens to the search region while evolving trajectory and appearance. In order to demonstrate the versatility of our model, we challenge it with complex scenarios that pose significant tracking difficulties, including appearance variation, partial occlusion, distribution, and motion blur, as shown in Figure 5. Perceptibly, our model adeptly captures the appearance evolution in successive frames within each of these challenging scenarios.

When confronted with rapid changes in target appearance and partial occlusions, traditional trackers tend to respond inadequately to these short-term variations. Furthermore, incorrect updates to the target's appearance in such scenarios can render the tracker agnostic to the target in subsequent frames. Our approach leverages the mutual complementation of both appearance and trajectory evolution, consecutively pinpointing the object's location. Joint evolution constructs a more comprehensive representation of the target, thus strengthening inter-frame autoregression in scenarios with incoherent visual or motion cues.

sequence augmentation	AO	$SR_{0.5}$	$SR_{0.75}$
fixed interval	74.8	84.6	71.3
random interval	75.1	84.9	70.9
reverse video	**75.9**	**85.4**	**72.7**

Table 6. **Sequence augmentation comparison**.

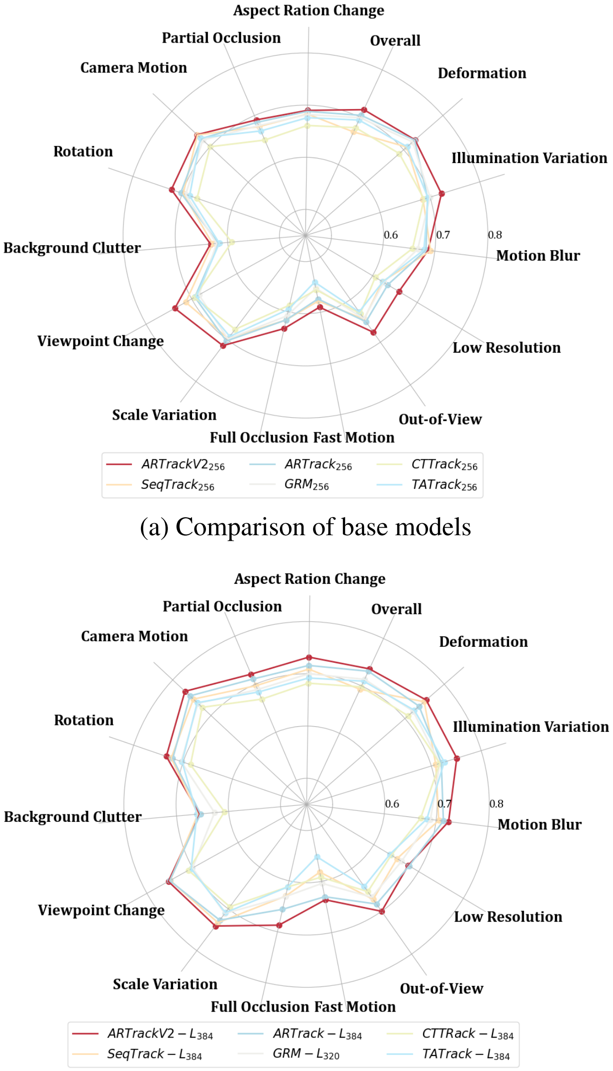
Sequence Augmentation. Sequence-level data augmentation [19, 67] is a widely used technique in video tasks, but it is rarely employed in visual tracking due to the prevalent use of frame-level training. In contrast, we embrace sequence-level training, where models are trained using video clips instead of image pairs. Therefore, it is necessary to explore video data augmentation, as demonstrated in Table 6. We experimented with sampling the video at fixed or random intervals to augment the training data. Unfortunately, both approaches led to a decrease in precision as they disrupted temporal continuity. In contrast, reverse augmentation, which simply plays the video backward, maintains the data distribution well. This straightforward method increases the AO in GOT-10k by 0.7%.

5. Conclusion

We introduce ARTrackV2, an end-to-end tracker that extends the concepts of the predecessor by implementing a unified generative framework that jointly evolves trajectory and reconstructs appearance. In continuous time-series, ARTrackV2 simultaneously "reads out" the target's localization and "retells" appearance variation. Then propagating and forecasting trajectory-appearance prompts into successive frames strengthens inter-frame autoregression. Moreover, we employ a pure encoder architecture that enables parallel processing of all tokens within a frame and eliminates less efficient intra-frame autoregression. ARTrackV2 demonstrates remarkable advancements in both performance and efficiency. In the future, we consider extending this unified framework to encompass multiple video-related tasks.

Appendix

This appendix encompasses comprehensive attribute results from the LaSOT [22] datasets. Additionally, it delineates the performance achieved on the TNL2K [59], NFS [34], and UAV123 [46] datasets. Subsequent sections detail extensive ablation experiments, meticulously examining the impacts of sequence format, trajectory-appearance evolution, and the masking ratio within the appearance model. Finally, it includes an array of rich visualizations, encompassing additional cross-attention maps and image reconstruction visualizations.



(a) Comparison of base models

(b) Comparison of large models

Figure 6. Comparison to other state of the art trackers, using the AUC score under different attributes of LaSOT [22] test set.

A. LaSOT Attributes Results

LaSOT [22] benchmark additionally offers a comprehensive range of specific attributes for each video sequence, such as Out-of-View, Full Occlusion, Viewpoint Change, and so on. We evaluate the AUC score under various attributes separately, presenting a comparative analysis between ARTrackV2 and other state-of-the-art trackers as shown in Figure 6(a) and Figure 6(b). ARTrackV2 demonstrates noteworthy advancements, particularly in handling scenarios involving Full Occlusion, Partial Occlusion, and Illumination Variation. These attributes necessitate continuous appearance evolution for robust and reliable tracking.

B. Performance on TNL2K [59], NFS [34] and UAV123 [46]

To showcase the robustness of our ARTrackV2 in a broader spectrum of scenarios, we conducted evaluations on three additional benchmarks: 1) TNL2K, is a multi-modal dataset that incorporates natural language tagging and comprises videos demonstrating pedestrian scenarios with cloth/face alterations; 2) NFS, is a dataset that relies on high-speed photography, offering videos at a higher frame rate (240 FPS); 3) UAV123, is a aerial dataset contains long-term videos in the complex scene. Our ARTrackV2-L_{384} consistently outperforms other trackers across these benchmarks, as illustrated in Figure 7. Notably, our models achieve this heightened performance without compromising on inference speed.

C. Sequence Format Study

sequence form	AO(%)	$SR_{0.5}$(%)	$SR_{0.75}$(%)
$[\boldsymbol{x}_{\min}, \boldsymbol{y}_{\min}, \boldsymbol{x}_{\max}, \boldsymbol{y}_{\max}]$	**75.9**	**85.4**	**72.7**
$[x, y, w, h]$	74.5	84.2	70.2
$[x, y, w, h]\dagger$	75.4	85.0	72.2
$[(x, y)_{t-l}, (x, y)_{b-r}]$	75.2	84.4	71.0

Table 7. **Sequence format study**.

Before where to look at the target, in which form (sequence format) to "read out" the target's position is still controversial within generative paradigm trackers. As Table 7 presented, we explore three sequence formats: 1) [$x_{\min}$, $y_{\min}$, $x_{\max}$, $y_{\max}$] leverages a set of top-left and bottom-right corner coordinates to denote target's localization. It utilizes a shared vocabulary to depict the token corresponding to the coordinate, akin to the approach adopted by ARTrack [60]. 2) $[x, y, w, h]$ denotes that generates the object's center point $[x, y]$ and scale $[w, h]$, employing a unified vocabulary that represents both position and height/width, as seen in the methodology by SeqTrack [11]. Moreover, we experiment with decoupling the position and scale using

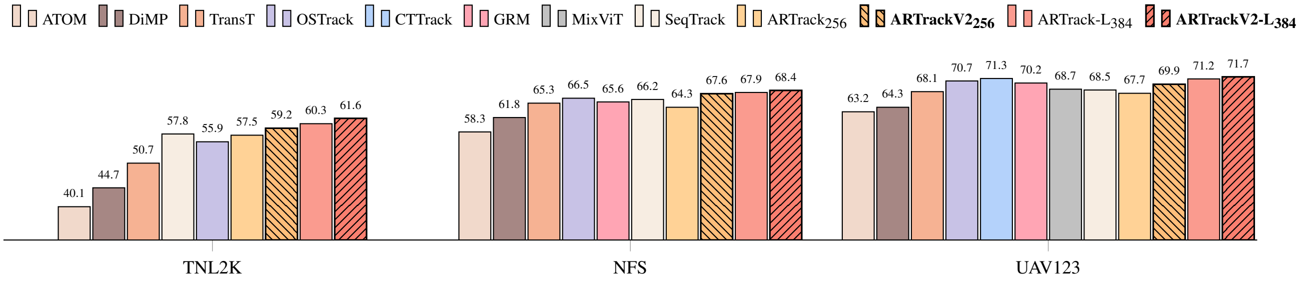


Figure 7. State-of-the-art comparison on TNL2K [59], NFS [34] and UAV123 [46].

separate vocabularies, denoted by †. 3) Drawing inspiration from Polyformer [43], we configure the sequence as [(x, y)t-l, (x, y)b-r], wherein "t-l" represents top-left and "b-r" represents bottom-right. This format includes two vertex coordinates, employing a 2D coordinate codebook. This codebook maps each floating-point coordinate to a bilinear interpolation of adjacent coordinates in 2D space without any quantization.

Our findings indicate that the $[x_{\text{min}}, y_{\text{min}}, x_{\text{max}}, y_{\text{max}}]$ format surpasses the others. This observation primarily stems from 1) A unified vocabulary, contrary to our default setting, induces confusion between position and scale, resulting in lower precision 2) The variant $[x, y, w, h]$†, which decouples the position and scale using separate vocabularies, demonstrates better performance compared to a unified vocabulary. However, the utilization of multiple vocabularies introduces the potential interference between them. 3) $[(x, y)_{\text{t-l}}, (x, y)_{\text{b-r}}]$, which implies the utilization of almost infinite large vocabulary, significantly slows down the training and impaired accuracy.

D. Trajectory and Appearance Evolution

	AO(%)	$SR_{0.5}$(%)	$SR_{0.75}$(%)	FPS
ARTrackV2$_{256}$	**75.9**	**85.4**	**72.7**	94
w/o trajectory	72.9	82.7	71.4	96
w/o appearance	72.3	81.9	70.3	116
w/o both	69.8	79.5	67.6	**120**

Table 8. **Trajectory-appearance evolution**.

The primary focus of our work lies in the evolution of both trajectory and appearance. Therefore, it becomes imperative to conduct comparative experiments to assess the individual impacts of respective. Table 8 illustrates the performance of these experiments, revealing that the exclusion of either trajectory or appearance evolution leads to a notable decline in performance. This performance degradation is attributed to the fact that individual evolution components are insufficient to describe coherent target variation, particularly in intricate tracking scenarios. This underscores the significance of jointly considering trajectory and appearance evolution for optimal tracking performance. Furthermore, our experiments involved the elimination of both trajectory and appearance evolution, resulting in a more pronounced decline in accuracy. In this variation, continuous tracking is transformed into a frame-level template-matching approach which breaks the time continue of tracking.

E. Masking Ratio for Appearance Modeling

ratio [%]	AO	$SR_{0.5}$	$SR_{0.75}$
0	74.7	84.0	70.8
25	75.0	84.2	70.9
50	75.2	84.5	71.2
75	75.5	84.8	71.9
90	**75.9**	**85.4**	**72.7**
95	75.3	84.7	71.6

Table 9. **Masking ratio study**.

In ARTrackV2, a key aspect of the design is the incorporation of an exceptionally high masking ratio. During the training phase, we randomly mask appearance tokens after feeding them into the transformer encoder based on a predetermined masking ratio prior to target reconstruction. The influence of various masking ratios on model accuracy is presented in Table 9. Unlike BERT [18] and MAE [30], our findings indicate that the model accuracy consistently improves as the masking rate increases until reaching a masking ratio of 90%. We posit that this phenomenon may arise from the redundancy and relevance of temporal information in comparison to textual and image cues [53,57]. Consequently, even when subjected to exceptionally high occlusion ratios, ARTrackV2 exhibits the capability to generate rational outputs.

F. More Visualization of Appearance Tokens

Within this section, we introduce additional visualizations of the cross-attention map between appearance tokens and the search region. As depicted in Figure 8, the findings from these visual representations align with the previous spot: the

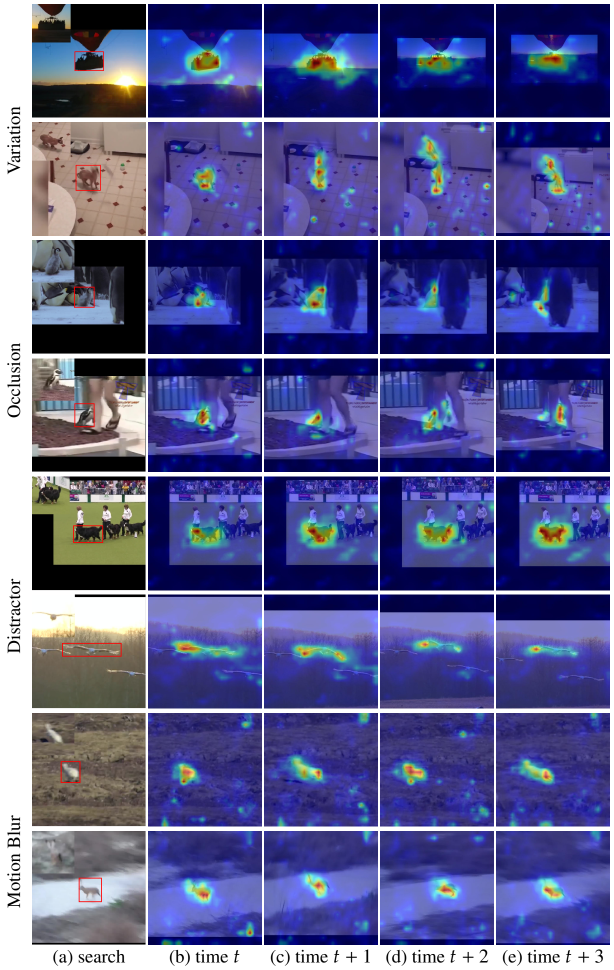


Figure 8. **More attention visualization**. (a): Search region and template. The red boxes denote the ground truth. (b)-(e): Appearance tokens to search the cross-attention map of ARTrackV2.

visualizations substantiate the adaptability and versatility of our model, particularly evident in challenging scenarios.

Additionally, our endeavor to visualize appearance tokens in an intuitive way encountered considerable challenges. We utilized dimensionality reduction techniques like t-distributed stochastic neighbor embedding (t-SNE) [54] to transform high-dimensional appearance tokens into one-dimensional spaces, aiming to produce grayscale images. Regrettably, the resolution of appearance tokens within the feature space remains notably limited. Consequently, the generated grayscale images are blurry with unclear boundaries due to this low-resolution representation.

G. Visualization of Image Reconstruction

To offer a more intuitive portrayal of our appearance evolution, distinct from visualizing the cross-attention map within the feature domain between appearance tokens and search, we conduct a quantitative analysis of appearance reconstruction within the image pixel domain. This analysis leverages an image reconstruction objective, aimed at reconstructing the target within the image pixel domain, as detailed in the ablation experiment. Illustrated in Figure 9, the first two rows depict tracking under appearance variation scenarios. ARTrackV2 adeptly captures long-term appearance evolution, ensuring accurate appearance descriptions even amidst significant changes, thereby facilitating precise tracking in complex environments. The subsequent two rows showcase tracking in occlusion scenarios. It's evident that when the target is unobstructed, ARTrackV2 accurately captures appearance changes. However, in instances where the target becomes occluded, the appearance token reconstructs the previous visible appearance. This prevents erroneous propagation of appearance, which could otherwise mislead

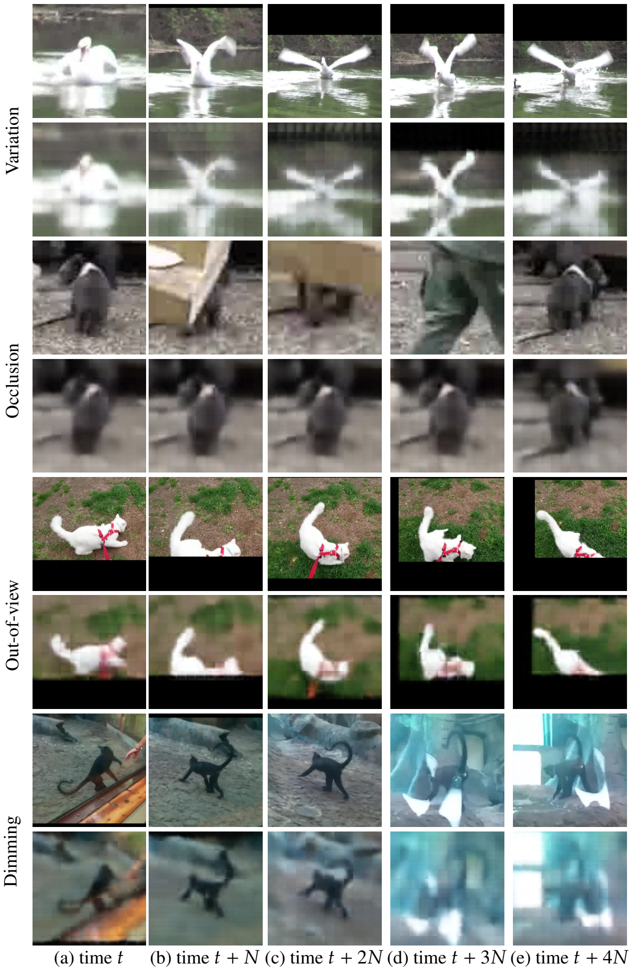


Figure 9. **Image reconstruction visualizations**. Reconstructed appearance in the image pixel domain of GOT-10k val set. For each video sequence, we show the target's appearance (top) and reconstructed appearance image (bottom). (a)-(e) represented sampling video frames at a fixed interval of N.

the model by providing nonsensical target localization. Upon the reappearance of the target, its appearance is accurately captured and faithfully reconstructed. The following two rows exemplify our tracker's proficiency in managing out-of-view targets. As the object gradually moves out of sight, our tracker adeptly adjusts to appearance changes, ensuring a seamless evolution in the reconstructed target's appearance. In the final two rows, the target's illumination gradually fluctuates. The appearance model notably captures these illumination variations with precision, resulting in a smoother and clearer evolution in appearance.

References

[1] Aiatrack: Attention in attention for transformer visual tracking. In ECCV, pages 146–164. Springer, 2022. 2

[2] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Transactions on Signal Processing, 50(2):174–188, 2002. 1

[3] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In ECCV, 2016. 2, 4, 6

[4] M Bertozzi, A Broggi, A Fascioli, A Tibaldi, R Chapuis, and F Chausse. Pedestrian localization and tracking system with kalman filtering. In IV, pages 584–589. IEEE, 2004. 1

[5] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In ICCV, pages 6182–6191, 2019. 2

[6] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In ECCV, 2018. 1

[7] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qiuhong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: A simplified architecture for visual object tracking. In ECCV, pages 375–392. Springer, 2022. 2

[8] SY Chen. Kalman filter for robot vision: a survey. IEEE Transactions on Industrial Electronics, 59(11):4409–4420, 2011. 1

[9] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. ICLR, 2021. 2, 3

[10] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey Hinton. A unified sequence interface for vision tasks. In NeurIPS, 2022. 3

[11] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In CVPR, pages 14572–14581, 2023. 2, 5, 6, 9

[12] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In CVPR, 2021. 2, 6

[13] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In CVPR, 2022. 2, 4, 6, 7, 8

[14] Yutao Cui, Tianhui Song, Gangshan Wu, and Limin Wang. Mixformerv2: Efficient fully transformer tracking. NeurIPS, 2023. 2, 7

[15] Kenan Dai, Yunhua Zhang, Dong Wang, Jianhua Li, Huchuan Lu, and Xiaoyun Yang. High-performance long-term tracking with meta-updater. In CVPR, 2020. 2, 8

[16] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In CVPR, 2019. 2

[17] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In CVPR, 2017. 6

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv, 2018. 10

[19] Shuangrui Ding, Maomao Li, Tianyu Yang, Rui Qian, Haohang Xu, Qingyi Chen, Jue Wang, and Hongkai Xiong. Motion-aware contrastive video representation learning via foreground-background merging. In CVPR, pages 9716–9726, 2022. 8

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In ICLR, 2021. 3

[21] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Mingzhen Huang, Juehuan Liu, Yong Xu, Chunyuan Liao, Lin Yuan, and Haibin Ling. Lasot: A high-quality large-scale single object tracking benchmark. International Journal of Computer Vision, 2021. 5, 6

[22] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In CVPR, 2019. 5, 6, 9

[23] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In CVPR, 2019. 1

[24] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In CVPR, pages 13774–13783, 2021. 2, 8

[25] Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In CVPR, pages 18686–18695, 2023. 5, 6

[26] Zhora Gevorgyan. Siou loss: More powerful learning for bounding box regression. arXiv, 2022. 3, 4

[27] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In CVPR, 2020. 2

[28] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In ICCV, Oct 2017. 8

[29] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and P-J Nordlund. Particle filters for positioning, navigation, and tracking. IEEE Transactions on Signal Processing, 50(2):425–437, 2002. 1

[30] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In CVPR, pages 16000–16009, 2022. 3, 4, 10

[31] Kaijie He, Canlong Zhang, Sheng Xie, Zhixin Li, and Zhiwen Wang. Target-aware tracking with long-term context attention. AAAI, 2023. 5, 6

[32] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019. 5, 6, 8

[33] Sajid Javed, Martin Danelljan, Fahad Shahbaz Khan, Muhammad Haris Khan, Michael Felsberg, and Jiri Matas. Visual object tracking with discriminative filters and siamese networks: a survey and outlook. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(5):6552–6574, 2022. 1

[34] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In ICCV, 2017. 9, 10

[35] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In ICCV, 2017. 1

[36] Minji Kim, Seungkwan Lee, Jungseul Ok, Bohyung Han, and Minsu Cho. Towards sequence-level training for visual tracking. In ECCV, 2022. 2, 4, 6

[37] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In CVPR, 2019. 2, 4

[38] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In CVPR, pages 8971–8980, 2018. 2

[39] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In CVPR, 2018. 1

[40] Peixia Li, Boyu Chen, Wanli Ouyang, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Gradnet: Gradient-guided network for visual object tracking. In ICCV, pages 6162–6171, 2019. 2

[41] Liting Lin, Heng Fan, Yong Xu, and Haibin Ling. Swintrack: A simple and strong baseline for transformer tracking. In NeurIPS, 2022. 2, 3, 6

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In ECCV, 2014. 5

[43] Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R Manmatha. Polyformer: Referring image segmentation as sequential polygon generation. In CVPR, pages 18653–18663, 2023. 10

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In ICLR, 2019. 5

[45] Alan Lukežic, Tomáš Vojír, Luka Cehovin Zajc, Jirí Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In CVPR, 2017. 6

[46] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In ECCV, 2016. 9, 10

[47] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In ECCV, 2018. 5, 6

[48] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In CVPR, 2016. 1

[49] Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. Motiontrack: Learning robust short-term and long-term motions for multi-object tracking. In CVPR, pages 17939–17948, 2023. 4

[50] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR, 2020. 2

[51] Zikai Song, Run Luo, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Compact transformer tracker with correlative masked modeling. AAAI, 2023. 5, 6

[52] Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Transformer tracking with cyclic shifting window attention. In CVPR, pages 8791–8800, 2022. 2

[53] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. NeurIPS, 35:10078–10093, 2022. 10

[54] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(11), 2008. 11

[55] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In CVPR, 2020. 4, 6

[56] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by instance detection: A meta-learning approach. In CVPR, pages 6288–6297, 2020. 2

[57] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In CVPR, pages 14549–14560, 2023. 10

[58] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In CVPR, 2021. 6

[59] Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In CVPR, 2021. 9, 10

[60] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In CVPR, pages 9697–9706, 2023. 2, 3, 4, 5, 6, 7, 9

[61] Xing Wei, Anjia Cao, Funing Yang, and Zhiheng Ma. Sparse parameterization for epitomic dataset distillation. In NeurIPS, 2023. 3

[62] Shiuh-Ku Weng, Chung-Ming Kuo, and Shu-Kang Tu. Video object tracking using adaptive kalman filter. Journal of Visual Communication and Image Representation, 17(6):1190–1208, 2006. 1

[63] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In ICCV, 2021. 2, 5, 6, 7, 8

[64] Tianyu Yang and Antoni B Chan. Learning dynamic memory networks for object tracking. In ECCV, pages 152–167, 2018. 2

[65] Tianyu Yang, Pengfei Xu, Runbo Hu, Hua Chai, and Antoni B Chan. Roam: Recurrently optimizing tracking model. In CVPR, pages 6718–6727, 2020. 2

[66] Botao Ye, Hong Chang, Bingpeng Ma, and Shiguang Shan. Joint feature learning and relation modeling for tracking: A one-stream framework. In ECCV, 2022. 2, 3, 5, 6

[67] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, and Jinhyung Kim. Videomix: Rethinking data augmentation for video classification. arXiv, 2020. 8

[68] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In ECCV, pages 659–675. Springer, 2022. 4

[69] Lichao Zhang, Abel Gonzalez-Garcia, Joost Van De Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In ICCV, pages 4010–4019, 2019. 2

[70] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In ICCV, October 2019. 8

[71] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In CVPR, pages 22056–22065, 2023. 4

[72] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In ECCV, 2020. 2, 6

[73] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In ECCV, 2018. 2