DiffusionTrack: Point Set Diffusion Model for Visual Object Tracking

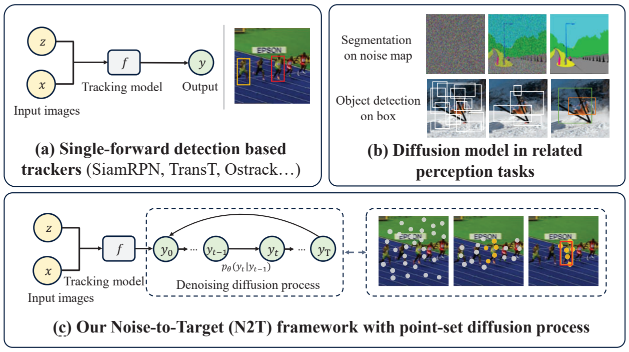
Fei Xie[†] Zhongdao Wang[‡] Chao Ma[†*]
[†] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[‡] Huawei Noah's Ark Lab
jaffe031@sjtu.edu.cn, wangzhongdao@huawei.com, chaoma@sjtu.edu.cn

Abstract

Existing Siamese or transformer trackers commonly pose visual object tracking as a one-shot detection problem, i.e., locating the target object in a ***single forward evaluation*** *scheme. Despite the demonstrated success, these trackers may easily drift towards distractors with similar appearance due to the single forward evaluation scheme lacking self-correction. To address this issue, we cast visual tracking as a point set based denoising diffusion process and propose a novel generative learning based tracker, dubbed DiffusionTrack. Our DiffusionTrack possesses two appealing properties: 1) It follows a novel noise-to-target tracking paradigm that leverages* ***multiple*** *denoising diffusion steps to localize the target in a dynamic searching manner per frame. 2) It models the diffusion process using a point set representation, which can better handle appearance variations for more precise localization. One side benefit is that DiffusionTrack greatly simplifies the post-processing, e.g. removing window penalty scheme. Without bells and whistles, our DiffusionTrack achieves leading performance over the state-of-the-art trackers and runs in real-time. The code is in* `https://github.com/VISION-SJTU/DiffusionTrack`.

1. Introduction

Visual Object Tracking (VOT) is one of the most fundamental computer vision problems with numerous applications. Recent prevalent Siamese [5, 31, 51, 105, 117] and transformer [14, 17, 25, 106, 113] based tracking approaches typically formulate visual tracking as a one-shot detection problem with a *single forward evaluation* scheme, i.e., these trackers first perform template-matching, and then predict the location and size changes of the target in a single forward pass. Despite the demonstrated success, regarding tracking as one-shot detection raises two critical issues: 1) Detectors emphasize the category-level difference to detect all potential objects from the background, while trackers focus on the instance-level difference to distinguish the target from distractors with a similar appearance. 2) It is difficult for trackers to localize targets undergoing large appearance variations and in complex scenarios in a single forward pass. Even humans can be easily confused by similar distractor objects at first glance.



(a) Single-forward detection based trackers (SiamRPN, TransT, Ostrack…)

(b) Diffusion model in related perception tasks

(c) Our Noise-to-Target (N2T) framework with point-set diffusion process

Figure 1. Compared to detection-based trackers with a single forward evaluation scheme (a), *e.g*. SiamRPN [51], TransT [14], OStrack [113], our proposed DiffusionTrack (c) localizes the target in a progressive diffusion manner. Furthermore, we tailor a point set representation to model the target object for the tracking task, instead of adopting existing representations such as noise maps for segmentation [13, 42] and random boxes for object detection [11].Better viewed with zoomed in.

To address these issues, we reformulate the tracking problem as a Noise-to-Target (N2T) process shown in Fig. 1, which explicitly mimics the coarse-to-fine searching mechanism of human vision. Our intuition is that we should empower a tracker with the ability of self-correction, by which it can progressively differentiate the target and fully exploit the rich contexts from the background and distractors. Such a progressive searching manner is intuitively superior to predicting the target size and location changes in merely one forward pass. To this end, we construct a denoising diffusion process, originally proposed for generative image tasks [20, 73, 79], to infer the target from random hypotheses on the entire frame step by step (see

[*] Corresponding author.

Fig. 1 (c)). In contrast to single forward detection based trackers [14, 51, 113], our prediction stage progressively refines the target hypotheses described by point sets through multiple diffusion denoising steps, taking advantage of self-correction to facilitate better differentiation of the target from distractors.

In this work, we propose a novel tracking framework dubbed DiffusionTrack. DiffusionTrack has an encoder-decoder architecture, where the encoder extracts target-aware features and the decoder predicts the target in a denoising diffusion process. Our decoder is a stack of multiple diffusion layers which can refine target estimations sequentially. The detailed structure of the diffusion layer is presented in Sec. 3.2. DiffusionTrack tackles the VOT task with a diffusion model by casting object tracking as a generative task over the space of point sets in the search region. We decouple the training and inference stages: 1) At the training stage, Gaussian noise is added to the ground truths to obtain noisy point-set estimations. Then decoder is trained to predict the ground truth without noise. 2) At the inference stage, DiffusionTrack estimates the target by reversing the learned diffusion process, which refines random point sets to focus on the target. Notably, our DiffusionTrack has two appealing merits: it uses an arbitrary number of points to represent the target and arbitrary steps to filter out background clutter in each frame. Thus, DiffusionTrack can handle challenging scenarios and achieve a dynamic trade-off between efficiency and effectiveness.

Extensive experiments on large-scale VOT benchmarks show that our proposed DiffusionTrack outperforms recent state-of-the-art trackers. For instance, under fair conditions, DiffusionTrack-B256 obtains a 75.2% AO score on the GOT-10k [39] dataset, surpassing OSTrack-256 [113] by 4.2% and SeqTrack-B256 [15] by 1.0%. In summary, the contributions of this work are as follows:

- We reformulate the tracking problem as a Noise-to-Target process and are the first to adopt the diffusion model to progressively predict the target per frame.
- We make the first attempt at introducing point-set representation into the denoising diffusion model, which can better handle appearance deformation and occlusions in complex tracking scenarios.
- Our proposed DiffusionTrack method achieves state-of-the-art results on four large-scale VOT benchmarks.

2. Related Work

Diffusion model. Diffusion models have recently demonstrated remarkable results in fields including computer vision [3, 24, 30, 32, 38, 69, 73, 77, 78, 81, 110, 116], nature language processing [2, 28, 55], audio processing [40, 45, 50, 72, 89, 102, 109] and graph-related topics [41]. In light of the great achievements of generative models, recent works [11, 42] demonstrate the potential of diffusion models for discriminative perception tasks. Some pioneering works attempt to adopt the diffusion model for image segmentation [1, 4, 7, 13, 29, 43, 99]. For example, [13] adopts the Bit Diffusion model [12] for panoptic segmentation [46]. DDP [42] concatenates a noise map with a feature map, then input them into the decoder to predict masks for semantic segmentation using the diffusion process. DiffusionDet [42] successfully applies the diffusion model to object detection by predicting bounding boxes from noisy boxes. Inspired by the success of diffusion models, in this work, we develop a point set based diffusion model to facilitate visual object tracking.

Visual object tracking. Siamese-based [16, 51, 52, 105, 117] and transformer-based [14, 17, 25, 56, 106, 113] trackers, which have attained great attention for their dominant performance and speed, share a bunch of similarities with object detection methods [9, 74, 75, 90]. The pioneering tracker SiamRPN [51] and its follow-up works [14, 16, 17, 25, 52, 56, 105, 106, 113, 117, 118] formulate visual tracking as a one-shot detection problem with a single forward pass evaluation scheme. Advanced by modern visual foundation models [21, 33, 49, 58, 88, 97], prediction design [35, 54, 75, 90, 120] and transformer models [21, 58, 70, 91, 97, 100], one-shot detection trackers have achieved great success, but they may easily drift towards distractors due to the single forward pass evaluation scheme lacking self-correction. To address this, a larger number of trackers explicitly exploit dynamically optimized modules to handle challenging factors. Representative tracking methods include Discriminative Correlation Filter [6, 18, 19, 36, 60, 65], template update mechanism [26, 103, 106, 111, 115], model fine-tuning [53, 67, 94], cascade structure [22, 93, 107] and the recent autoregressive decoder [15, 104]. In contrast to these dynamically optimized modules, we reformulate visual tracking as a Noise-to-Target process and resort to diffusion models on point set to localize the target per frame progressively.

Point set object representation. Traditional detectors or tracking methods [51, 74, 75, 90, 105, 120] primarily represent objects using axis-aligned bounding boxes, which are convenient to annotate with little ambiguity. However, these methods may encounter difficulties detecting targets undergoing appearance deformation and severe object occlusions. The RPT [62] tracker is the pioneering work to model the target state with a set of representative points, which borrows the prediction head consisting of two-stage convolutional layers from the Reppoints detector [112]. Inspired by point set based approaches in vision tasks [62, 63, 112, 120], we adopt point set to model the denoising diffusion process, which can better handle challenging scenarios. The difference is that previous point set based detectors or trackers empirically design reference

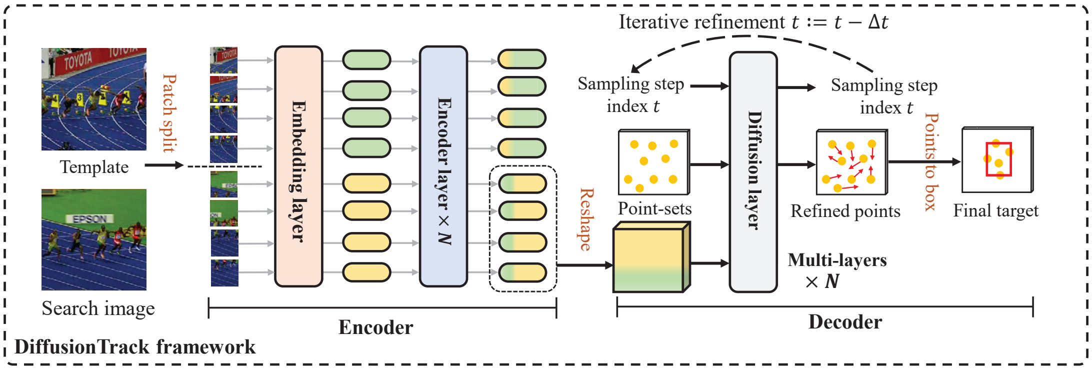


Figure 2. Architecture of DiffusionTrack. It has an encoder-decoder structure. The encoder extracts target-aware features and feeds search features into the decoder. The decoder, comprising of a stack of diffusion layers, refines the point set groups to localize the target.

points [112] in a per-pixel manner, while our point sets are arranged to sample the entire image region to discriminate the instance-level differences explicitly. To the best of our knowledge, no prior art has adopted point set to represent objects for diffusion models.

3. Method

In this section, we first introduce the preliminaries on visual tracking and diffusion model. Then, we present the proposed Noise-to-Target tracking paradigm and the model architecture of DiffusionTrack. Finally, we depict the training and inference details of DiffusionTrack.

3.1. Preliminaries

Visual tracking. Given state b_z of a target in template image z, the objective of visual tracking is to localize target b from the continuous input image x, where b denotes an axis-aligned bounding box. The mainstream tracking paradigm aims to learn a deep tracking model f such that:

$$b = f(z, x, b_z), \tag{1}$$

where the learned model f can predict the location and size changes of the target in a single forward evaluation, given an image pair (template z and search image x) as input at both the training and inference stages.

Diffusion model. Diffusion models [37, 82, 84, 85] are generative likelihood-based models inspired by nonequilibrium thermodynamics [85, 86]. These models define a Markovian chain of diffusion forward process by gradually adding noise $\boldsymbol{N_s}$ to sample data $\boldsymbol{i}$. The forward noise process is defined as:

$$q(\boldsymbol{i}_t|\boldsymbol{i}_0) = \mathcal{N}(\boldsymbol{i}_t|\sqrt{\bar{\alpha}_t}\boldsymbol{i}_0, (1-\bar{\alpha}_t)\boldsymbol{N_s}). \tag{2}$$

The forward process transforms data sample $\boldsymbol{i}_0$ into a latent noisy sample $\boldsymbol{i}_t$, $t \in \{0, 1, ..., T\}$, by adding noise to $\boldsymbol{i}_0$. $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s = \prod_{s=0}^{t}(1-\beta_s)$ and β_s represents the noise variance schedule [37]. During training, a model $f_\theta(\boldsymbol{i}_t, t)$ is trained to predict the true data sample $\boldsymbol{i}_0$ given a noisy sample $\boldsymbol{i}_t$ at diffusion step t by minimizing the ℓ_2 loss [37]:

$$\mathcal{L}_{\text{train}} = \frac{1}{2}||f_\theta(\boldsymbol{i}_t, t) - \boldsymbol{i}_0||^2. \tag{3}$$

At the inference stage, data sample $\boldsymbol{i}_0$ is predicted from noise $\boldsymbol{i}_T$ with model f_θ in an iterative manner [37, 84], *i.e.*, $\boldsymbol{i}_T \rightarrow \boldsymbol{i}_{T-\Delta} \rightarrow ... \rightarrow \boldsymbol{i}_0$, where Δ can be dynamically determined. More details can be found in the appendix.

3.2. DiffusionTrack framework

We first depict our Noise-to-Target scheme and then introduce the model architecture. The overall framework of DiffusionTrack is presented in Fig. 2. It consists of a transformer-based encoder and a diffusion-based decoder. The target state is described by point sets.

Noise-to-target tracking paradigm. We describe how the visual tracking problem is posed as a diffusion process. In the tracking process, we adopt a group of point sets G_N to estimate the target state b (location and size), where $G_N = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N\}$ and $\mathbf{p}$ is a group of point sets $\{(x_k, y_k)\}_{k=1}^{m}$. Each point set $\mathbf{p}$ extracts target proposal from search region. The goal of the Noise-to-Target paradigm is to learn a tracking model f which can gradually refine the target estimation through a total of T diffusion steps with an interval of ΔT:

$$G_N^T \xrightarrow{f} G_N^{T-\Delta T} \xrightarrow{f} ... \xrightarrow{f} G_N^0, \tag{4}$$

where diffusion step $T \rightarrow 0$ depicts the target estimation changes from an absolute random state to the highest certainty. Thus, the tracking process based on diffusion model f can be formulated as:

$$\begin{aligned} \Delta G_N^t, C_N^t &= f(z, x, t, G_N^{t-\Delta t}), \\ G_N^t &= G_N^{t-\Delta t} \oplus \Delta G_N^t, \end{aligned} \tag{5}$$

where model f refines the current estimation by knowing the diffusion step index t and the previous-step estimation

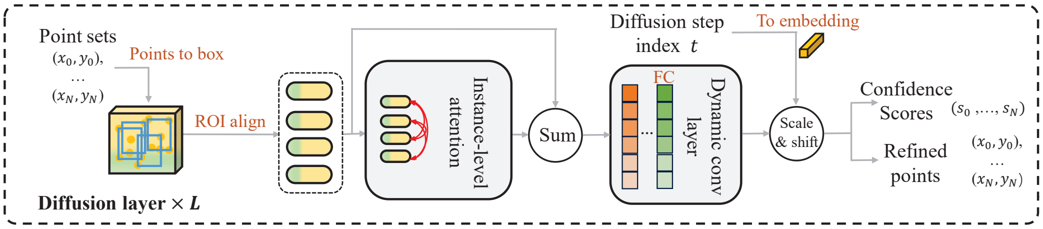


Figure 3. Details of a diffusion layer. It consists of three components: 1) Global instance layer: it produces target proposals in a generative style and models the instance-level relationship. 2) Dynamic conv layer: it performs dynamic convolution with instance features. 3) Refinement layer: it refines the point sets and estimates corresponding confidence scores.

$G_N^{t-\Delta t}$; $\oplus$ is element-wise summation. C_N^t is the corresponding confidence score for G_N^t. Then, the final target state b as tracking output can be obtained by:

$$\begin{aligned} B_N &= \Gamma(G_N^t), \\ b = B_{\hat{i}}, \quad &\text{s.t.}\hat{i} = \arg\max\{C_i\}_{i=1}^N, \end{aligned} \tag{6}$$

where Γ is a differentiable function that transforms point sets into box format. We adopt Min-Max [112] for the function Γ in this work. Please refer to the appendix for details.

Encoder. We adopt a plain vision transformer (ViT) [21] as the encoder and encode the search and template images jointly [15, 113]. The encoder extracts the features of the search and template images jointly and learns feature-level correspondence. We first split the template and search images into patches $\{z_{Nz}, x_{Nx}\}$, before projecting them into feature tokens:

$$[f_z, f_x] = [\mathbf{E}x_1, \ldots, \mathbf{E}x_{N_x}, \mathbf{E}z_1, \ldots, \mathbf{E}z_{N_z}], \tag{7}$$

where $\mathbf{E}$ is a linear projection layer. The features $\{f_z, f_x\}$ are encoded in a joint style:

$$\begin{aligned} f_{zx}^i &= \text{Concat}(f_z^i, f_x^i), \quad i \in \{0\} \\ f_{zx}^i &= \text{BLK}^i(f_{zx}^{i-1}), \quad i \in \{1, l\} \end{aligned} \tag{8}$$

where $\{f_z, f_x\}$ are first concatenated at the beginning and jointly encoded in the multiple transformer blocks $\{\text{BLK}^i, 1 \leq i \leq l\}$. Only the features of the search image f_x are fed into the decoder. Here, we omit position encoding for simplicity.

Decoder. The decoder of DiffusionTrack is a stack of diffusion layers. As shown in Fig. 3, each diffusion layer consists of three components: global instance interaction, a dynamic convolutional layer, and a refinement layer. Each diffusion layer takes a total of N point sets $G_N^{t-\Delta t}$ from previous diffusion step $t - \Delta t$ to crop instance features $F_N^{\text{ins}} = \{\mathbf{f}_0^{\text{ins}}, \mathbf{f}_1^{\text{ins}}, ..., \mathbf{f}_N^{\text{ins}}\}$ through RoI pooling [34, 75], after which the global relationships are modeled by a Simplified Self-Attention (SSA) layer:

$$\begin{aligned} F_N^{\text{ins}} &= \text{RoI}[f_x, \Gamma(G_N^{t-\Delta t})], \\ F_N^{\text{ins}} &:= \text{SSA}(F_N^{\text{ins}}), \end{aligned} \tag{9}$$

where F_N^{ins} includes N instance-level embeddings, and the self-attention layer removes the linear projection layer in the original Multi-head Attention layer [21, 91]. In the refinement layer, we embed the diffusion step information into the instance embedding and predict the target state through the dynamic convolutional layer:

$$\begin{aligned} F_N^{\text{ins}} &:= \text{DyConv}[\text{GAP}(F_N^{\text{ins}}), F_N^{\text{ins}}], \\ F_N^{\text{ins}} &:= F_N^{\text{ins}} \oplus \text{ToEmbed}(t), \end{aligned} \tag{10}$$

where GAP and $\oplus$ denote global average pooling and element-wise summation, respectively; ToEmbed is an embedding network that transforms a step index t from scalar into a feature vector; DyConv$[A, B]$ denotes convolution between input B and convolution kernel A. Eventually, we predict the refinement and confidence scores of point-sets:

$$\Delta G_N^t = \phi_{reg}(F_N^{\text{ins}}), \quad C_N^t = \phi_{cls}(F_N^{\text{ins}}) \tag{11}$$

where $\{\phi_{reg}, \phi_{cls}\}$ are two light-weight convolutional networks; ΔG_N^t denotes the relative offsets to refine G_N^t and C_N^t is the confidence score. The outputs can be used for the next layer or iterative evaluation. G_N^t is transformed to box format for training supervision.

The difference between our decoder and that of DiffusionDet [11] is as follows: 1) DiffusionDet [11] predicts the category label for object detection, while our decoder only needs to perform binary classification. 2) DiffusionDet [11] has fixed layers to formulate the decoder, while our decoder can exit at early layers for speed-up (see Sec. 3.4). 3) Our decoder refines the point sets to gradually localize the target, while DiffusionDet [11] predicts the bounding boxes and category labels of objects in the image.

3.3. Training

In this section, we describe the process of training that train the tracking model f to predict the ground truth of target state b from random noisy conditions. The training procedure for DiffusionTrack is shown in Algorithm 1. During training, the tracking model f is trained to predict ground truth b_0 given its noisy version b_t sampled from a diffusion process $q(b_0|b_t)$ at diffusion step t. Each noisy version b_t at diffusion step t models the tracking estimation under dynamic tracking scenarios.

Algorithm 1 Training algorithm

```
Input: image pair $\{z, x\}$, GT box $b$.
Output: training loss
Extract search features $f_x$ ( Eq. 7 and Eq. 8).
Initialize the target estimations: $(N-1)$ point sets sampled from random distribution. $G_{N-1}^{Ne} \leftarrow \textbf{Rand}(N-1)$
Combine noisy target estimations with GT box $b$. $G_{N}^{Ne} \leftarrow \textbf{Concat}(\textbf{Box2Point}(b), G_{N-1}^{Ne}$
Construct noise signal and choose step index $t$. $t \leftarrow \textbf{Randint}(0, T), Ne \leftarrow \textbf{Randn}(\text{mean} = 0, \text{std} = 1)$
Signal scaling. $G_{N}^{Ne} \leftarrow \textbf{Norm}(G_{N}^{Ne})$
Corrupt target estimation $G_{N}^{Ne}$ input with noise and step index $t$. $GN_{N}^{t} \leftarrow \textbf{Schedule}(t) \times G_{N}^{Ne} + (1 - \textbf{Schedule}(t)) \times Ne$
Predict results $b_{\textbf{pred}}$ using decoder ( Eq. 5 and Eq. 6).
Obtain the training loss $\mathcal{L}$ (Eq. 12).
```

Point sets with random noise. We combine noisy point sets Ne and the ground truth b together to construct the input GN_{N}^{t} for the training stage. The noise scale for each time step t is controlled by a pre-defined monotonically decreasing schedule proposed in [12, 68]. Another hyper-parameter is the scale ratio between ground truth and noise (Signal-to-Noise Ratio). We empirically find that the Gaussian noise and setting larger SNR ($=1$) work best for diffusion-based tracking. More discussions can be found in Sec. 4.3 and the appendix.

Training loss. The diffusion decoder takes noisy point sets G_{N}^{t} as input, and predicts N confidence scores as well as relative movement of corresponding points. We apply the set prediction loss on the set of N predictions and assign multiple predictions to ground truth by selecting the top K ($=5$) predictions according to the IoU [114] scores. The overall loss function of DiffusionTrack is as follows:

$$\mathcal{L} = \sum_{i=1}^{L} (L_{cls}^{i} + \lambda_{iou} L_{iou}^{i} + \lambda_{L_1} L_{1}^{i}). \tag{12}$$

where L_{cls} is the weighted focal loss [54] for classification, L_1 and generalized IoU loss [76] L_{iou} are employed for box regression, λ_{iou} and λ_{L_1} are the weighting values. We adopt intermediate supervision [11] for total L diffusion layers.

3.4. Inference

The inference procedure of DiffusionTrack is a denoising sampling process from noise to target. Starting from random point set estimations, the tracking model progressively refines point sets to focus on the target, as shown in Algorithm 2. We slightly abuse of notations for clarity.

Dynamic inference. We introduce two dynamic settings of DiffusionTrack during inference: the first is the arbitrary number of target estimations N, and the second is the number of iterations τ. In contrast, previous detection-based trackers only predict the target in a single forward evaluation. For each evaluation step, previous target prediction $G_{N}^{t|(\tau-1)}$ is sent to DDIM [83] for producing the noisy version $G_{N}^{t|\tau}$ input for the next step. It is feasible to send the predicted boxes $G_{N}^{t|(\tau-1)}$ without DDIM [83] to the next step. But doing so neglects multiple diffusion step training, leading to sub-optimal results.

Algorithm 2 Inference algorithm (decoder only)

```
Input: target-aware search features $f_x$, decoder with $L$ diffusion layers $\textbf{Decoder}^{L}$, evaluation step $\tau$, total diffusion steps $T$.
Output: final prediction result $b^{\tau}$
Initialize target estimation $G_{N}^{1|1}$ from random distribution.
for $i = 1, 2, ..., \tau$ do
    Obtain diffusion step index $t$ for current evaluation ( Eq. 4). $t \leftarrow i * \Delta\tau, \Delta\tau \leftarrow T/\tau$
    if $i > 1$ then
        Replace current estimations with previous high-scoring point sets. $G_{N}^{t|i} \leftarrow \textbf{Renew}(G_{N}^{t|i}, G_{N}^{(t-\Delta t)|(i-\Delta\tau)})$
    end if
    Predict results from all $L$ diffusion layers (Eq. 5). $\{G_{N}^{t|i}, C_{N}^{t|i)}\}_{l=1}^{L} \leftarrow \textbf{Decoder}^{L}(f_x, i, G_{N}^{t|i})$
    Vote results. $b^{i} \leftarrow \textbf{Point2Box}(\textbf{Vote}(\{G_{N}^{t}, C_{N}^{t}\}_{l=1}^{L}))$
    Refine target estimation via DDIM for the next evaluation. $G_{N}^{(t+\Delta t)|(i+1)} \leftarrow \textbf{DDIM}(G_{N}^{t|i}, t, \Delta t)$
end for
```

Ensemble prediction. During the inference of each decoder layer and every evaluation step, the predicted boxes can be coarsely categorized into two types: predictions with high confidence scores and those with low scores. The high-scoring predictions are properly localized at the corresponding target, while the low-scoring ones mostly focus on the background. Motivated by these observations, we adopt the renewal strategy to replace these low-scoring estimations with random point sets and revive the high-scoring predictions, thus promoting the prediction in the next layer/step. Moreover, for multiple high-scoring estimations, we use the voting strategy to filter out the distractor objects so that the target can be localized more precisely.

Early exit. As each layer of the proposed decoder can generate predictions independently, the model inference of DiffusionTrack can exit at an early stage for speed-up. A simple threshold strategy can be applied to stop the inference when the maximum confidence score is larger than a predefined threshold value. More details are in Sec. 4.3.

4. Experiments

In this section, we first describe the implementation details. We then present extensive ablation studies of our method and compare it with SOTA trackers. Besides, we also showcase qualitative results and discuss the limitations of the proposed method.

Table 1. State-of-the-art comparisons on four large-scale benchmarks. We add a symbol * over GOT-10k to indicate that the corresponding models are only trained with the GOT-10k training set. Otherwise, the models follow the full-dataset training presented in Sec. 4.1. N2T denotes our Noise-to-Target tracking framework. (1)&(2) denotes the number of iterative evaluation. The top three results are highlighted with red, blue and green fonts, respectively.

	Method	LaSOT [23]			TNL2K [98]			TrackingNet [66]			GOT-10k* [39]		
		AUC	P_{Norm}	P	AUC	P_{Norm}	P	AUC	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$
N2T	DiffusionTrack-B256 (1)	70.8	79.8	76.7	56.4	72.5	57.3	83.8	88.2	82.1	74.8	85.4	72.0
	DiffusionTrack-B256 (2)	70.7	80.0	77.3	56.5	72.6	57.3	83.6	88.1	82.0	75.2	85.9	72.0
	DiffusionTrack-L256 (1)	72.3	81.8	79.1	56.8	72.8	57.7	85.2	89.6	84.8	74.7	85.6	71.8
Detection-Based	GRM-256 [27]	69.9	79.3	75.8	-	-	-	84.0	88.7	83.3	73.4	82.9	70.4
	ROMTrack-B256 [8]	69.3	78.8	75.6	-	-	-	83.6	88.4	82.7	72.9	82.9	70.2
	OSTrack-B256 [113]	69.1	78.7	75.2	55.9	-	-	83.1	87.8	82.0	71.0	80.4	68.2
	SimTrack-B224 [10]	69.3	78.5	-	55.6	-	-	82.3	86.5	-	68.6	78.9	62.4
	Mixformer-22k [17]	69.2	78.7	74.7	-	-	-	83.1	88.1	81.6	70.7	80.0	67.8
	AiATrack	69.0	79.4	73.8	-	-	-	82.7	87.8	80.4	69.6	63.2	80.0
	UTT [61]	64.6	-	67.2	-	-	-	79.7	-	77.0	67.2	76.3	60.5
	CSWinTT [87]	66.2	75.2	70.9	-	-	-	81.9	86.7	79.5	69.4	78.9	65.4
	STARK [106]	67.1	77.0	-	-	-	-	82.0	86.9	-	68.8	78.1	64.1
	SwinTrack-224 [56]	67.2	-	70.8	-	-	-	81.1	-	78.4	71.3	81.9	64.5
	RTS [71]	69.7	76.2	73.7	-	-	-	81.6	86.0	79.4	-	-	-
	Unicorn [108]	68.5	-	-	-	-	-	83.0	86.4	82.2	-	-	-
	SLT [44]	66.8	75.5	-	-	-	-	82.8	87.5	81.4	67.5	76.5	60.3
	SBT [25]	66.7	-	71.1	-	-	-	-	-	-	70.4	80.8	64.7
	TransT [14]	64.9	73.8	69.0	50.7	-	-	81.4	86.7	80.3	67.1	76.8	60.9
	SiamAttn [96]	56.0	64.8	-	-	-	-	75.2	81.7	-	-	-	-
	SiamBAN [16]	51.4	59.8	-	40.0	-	41.7	-	-	-	-	-	-
	SiamRPN++ [52]	49.6	56.9	49.1	41.3	-	41.2	73.3	80.0	69.4	51.7	61.6	32.5
Dynamically Optimized	SeqTrack-B256 [15]	69.9	79.7	76.3	54.9	-	-	83.3	88.3	82.2	74.7	84.7	71.8
	KeepTrack [64]	67.1	77.2	70.2	-	-	-	-	-	-	-	-	-
	AutoMatch [119]	58.3	-	59.9	-	-	-	76.0	-	72.6	65.2	76.6	54.3
	TrDiMP [95]	63.9	-	61.4	-	-	-	78.4	83.3	73.1	68.8	80.5	59.7
	ToMP [65]	68.5	79.2	73.5	-	-	-	81.5	86.4	78.9	-	-	-
	DSTrpn [80]	43.4	54.4	-	-	-	-	64.9	-	58.9	-	-	-
	Ocean [118]	56.0	65.1	56.6	38.4	-	-	-	-	-	61.1	72.1	47.3
	SiamR-CNN [92]	64.8	72.2	-	-	-	-	81.2	85.4	80.0	64.9	72.8	59.7
	DiMP [6]	56.9	65.0	56.7	-	-	-	74.0	80.1	68.7	61.1	71.7	49.2
	ATOM [19]	51.5	57.6	50.5	40.1	-	39.2	70.3	77.1	64.8	55.6	63.4	40.2

4.1. Implementation Details

Network architecture and training. Our DiffusionTrack adopts a ViT-Base [21] as the encoder with pretrained weights from DropTrack [101]. More details about the encoder are in the appendix. We use COCO [57], LaSOT [23], GOT-10k [39] and TrackingNet [66] as the training datasets. The total batch size for training on 4 NVIDIA A800 GPUs is set to 64. The template and search image are both cropped as 256×256. The total number of diffusion steps and target estimations are set to 1000 and 50, respectively. We train the model with AdamW [59] optimizer, set the initial learning rate for the backbone to 4×10^{-5} and other parameters to 4×10^{-4}. The total number of training epochs is set to 300 with 60k image pairs per epoch. The learning rate is decreased by a factor of 10 after 240 epochs. Other settings follow [15], and more details are in the appendix.

Online inference. During inference, each diffusion layer produces a 5D vector $T_i = (c, x1, y1, x2, y2)$ for each target estimation g_i^l, where c represents the confidence score of target classification, and $(x1, y1, x2, y2)$ denotes the point set. Afterward, we use NMS [75] processing to filter out aggregated prediction results from all layers and choose the target estimation with maximum confidence score.

4.2. Comparison to state-of-the-arts

We compare our proposed DiffusionTrack with SOTA trackers on four large-scale VOT benchmarks and one challenging VOT dataset. More results are provided in the appendix.

GOT-10k. GOT-10k [39] is a recent large-scale dataset containing over 10k videos for training and 180 for testing. GOT-10k has a zero overlap of object classes between the training and testing subsets. We strictly follow the official protocol, which forbids external datasets for training. In Tab. 1, under similar conditions, our base model with two iterative steps obtains a 75.2% AO score, outperforming exiting SOTA trackers which belong to the other two categories by a significant margin. DiffusionTrack also ranks first on the other two metrics: 85.9% in $SR_{0.5}$ and 72.0% in $SR_{0.75}$.

LaSOT. LaSOT [23] is a large-scale dataset with high-quality box annotations, and its testing subset has 280 videos with an average video length of 2448 frames. As shown in Tab. 1, DiffusionTrack-L256 achieves the top-rank AUC score (72.3%) and Precision score (79.1%), surpassing the SOTA trackers SeqTrack [15]/GRM [27]/ROMTrack [8] by 2.4/2.4/3.0 points in terms of AUC score. Under fair conditions,

Number	AO	SR$_{75}$	SR$_{50}$
2	**74.4**	**85.4**	**71.6**
3	74.2	85.1	71.2
5	73.4	83.9	70.9
7	73.7	84.6	70.8

(a) **Point set group**. Setting 2 points works best.

Scale	AO	SR$_{75}$	SR$_{50}$
0.1	68.5	78.8	63.9
1.0	**74.4**	**85.4**	**71.6**
2.0	72.8	82.9	70.2
3.0	71.9	81.7	68.1

(b) **Scaling factor**. The best scaling factor is 1.

Scale	AO	SR$_{75}$	SR$_{50}$
Uniform	72.6	83.5	69.8
Gaussian	**73.9**	**84.5**	**70.9**
Origin	71.0	82.1	68.3

(c) **Noise schedule**. Gaussian noise works best.

Depth	AO	SR$_{75}$	SR$_{50}$	Speed
1	71.0	81.1	66.8	45
2	72.2	82.3	69.8	42
4	73.0	83.3	70.1	38
6	**74.3**	**85.8**	**72.1**	**31**
12	71.4	81.7	67.8	25

(d) **Decoder depth** L. Six layers work best.

Thres.	AO	SR$_{75}$	SR$_{50}$	Speed
0.5	72.3	84.0	69.5	38
0.6	71.7	83.0	68.9	37
0.7	73.0	84.1	70.3	36
0.8	**73.7**	**84.3**	**70.5**	**35**
0.9	72.9	83.6	70.3	33
default	74.1	84.6	71.3	30

(e) **Early exit**. Threshold = 0.8 achieves the best AO.

Table 2. Ablation experiments with DiffusionTrack on GOT-10k [39] test set. We report the performance with 1 iteration step in (a), (b), (c), (d), and (e). If not specified, the default settings are: the number of points per group is 2; the scaling factor is set to 1; the noise schedule is Gaussian noise; the decoder has a depth of 6; training and test settings follow the official GOT-10k [39] protocol.

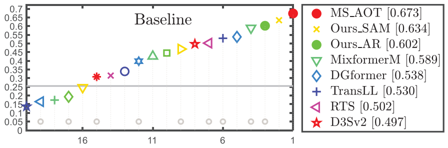


Figure 4. VOT2022 [48] results of our DiffusionTrack-B256(1) with Alpha-Refine [107] and SAM [47] model.

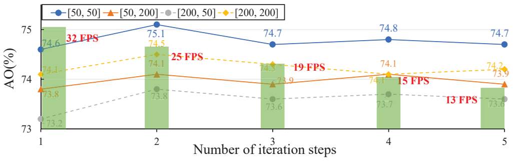


Figure 5. Ablations on the evaluation steps and the target proposals between training and testing. $[N_{train}, N_{test}]$ refer to the number of proposals for the two stages. Green bins denote speed.

DiffusionTrack-B256 (2) also achieves a better AUC score (70.7%) than OStrack [113] and SimTrack [10].

TrackingNet. TrackingNet [66] is a recent large-scale tracking benchmark consisting of 511 sequences for testing. The evaluation is performed on the online server. Tab. 1 shows that, compared with SOTA models, DiffusionTrack-L256 ranks first with an AUC score of 85.2% and a normalized precision of 89.6%.

TNL2k. TNL2k [98] is a recently released large-scale dataset with 700 challenging video sequences. The test subset of TNL2k contains 1598 video tests which can extensively evaluate the trackers. The results in Tab. 1 show that our DiffusionTrack-L256 surpasses all other trackers by a large margin and achieves the top-ranked performance of 56.8% AUC, outperforming Seqtrack by 1.9%. Our base model DiffusionTrack-B256 (1) also outperforms previous trackers by a notable margin on all three metrics under fair comparisons, *i.e.* equal image resolution and model size.

VOT2022. VOT2022 [48] is a challenging benchmark, which presents new video sequences yearly. When equipped with Alpha-Refine and SAM model, our method achieves competitive results as shown in Tab. 4.

4.3. Ablation Studies

We conduct detailed ablation studies with an extensive analysis of the effectiveness of our DiffusionTrack.

Point set group. Point sets are vital to generating target estimations to localize the target. We first investigate the impact of different numbers of points per group n. In Tab. 2a, $n = 2$ has almost the same performance (74.4 vs.74.2) as $n = 3$, and further increases in the number of points ($n = \{3, 5, 7\}$) leads to degraded performance. We infer that the degeneration may be caused by a lack of effective supervision for an excessive amount of points during training. Thus, we adopt $n = 2$ for a trade-off between efficiency and effectiveness.

Signal scale. The signal scaling factor controls the ratio between random noisy estimations and ground truths in the diffusion process. In Tab. 2b, we identify one as the optimal scaling factor. When the scaling factor is too small, *i.e.*, 0.1, we observe a significant performance drop (74.4 to 68.5), indicating that the ground truth is easily overwhelmed by a large magnitude of noise. However, further raising the scaling factor does not bring continuous performance gains. We argue that an excessive magnitude of the ground truth signal may encourage the model to learn a shortcut during training, thereby undermining its generalization ability.

Noise schedule. We compare the effectiveness of different noise schedules for DiffusionTrack in Tab. 2c. Compared to Uniform and Origin noise cases, the model using the Gaussian random noise schedule achieves notably better performance (73.9 vs. 72.6 vs. 71.0). This is attributed to Gaussian noise's mechanism of simulating the distribution of the target location in realistic scenarios, which prompts the tracking model to learn stronger denoising capabilities. More details can be found in the appendix.

Decoder depth. We study the effect of decoder depth in Tab. 2d and observe that a proper depth value is essential to the performance. The model performance improves as the depth increases within a reasonable range. Thus, we finally adopt a decoder with 6 layers.

Early exit. As each layer of the decoder can generate predictions independently, we show that DiffusionTrack can achieve a dynamic trade-off between accuracy and efficiency in Tab. 2e. We adopt a simple threshold strategy that

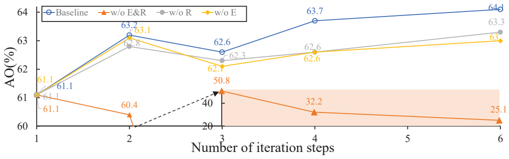


Figure 6. Ablations on the ensemble prediction strategy. “E” denotes estimations from all decoder layers that vote to the final result. “R” denotes high-scoring estimations from the previous layer, which replace the random target estimation for the next layer. The Baseline setting adopts two strategies. To prevent overfitting, we use the model checkpoint from the 50^{th} epoch for evaluation.

suspends the inference process when the maximum confidence score of the target estimation exceeds a pre-defined threshold value. DiffusionTrack can be 16.7% faster while only sacrificing 0.4% AO score. It enlightens us that a dynamic network architecture can exit early for speed-up when handling easy cases.

Multiple iterative evaluations. We further investigate the impact of varying numbers of evaluation steps, and corresponding results are illustrated in Fig. 5 and Fig. 6. Our findings indicate that DiffusionTrack exhibits consistent performance improvements as the number of iterations increases from 1 to 2. However, the performance gains almost saturate when the the number of evaluation steps is increased further. We suspect that the model is overfitting since the AO performance has already achieved 74%. Thus, in Fig. 6, we adopt a weak model which stops its training at an earlier stage. Substantial gains are observed even when the number of steps is increased to 6, *e.g.*, the Baseline case has its AO raised from 61.1% to 64.1% . These results validate the superiority of our progressive tracking paradigm, comparing to non-iterative detection based trackers. In Fig. 5, we also search for the optimal target estimation number ($= 50$) and find that the tracking model performs better when the estimation numbers $\{N_{train}, N_{test}\}$ in training and testing are consistent.

Ensemble prediction. We study the effects of the two proposed ensemble strategies, *i.e.*, voting and renewal, on boosting tracking performance. In Fig. 6, the model case deprived of both two strategies suffers from a dramatic performance drop (61.1 % to 25.1%), revealing that the model loses the ability to perform multiple evaluations. On the contrary, using either of the two strategies alone leads to consistent performance gains as the number of iteration steps increases. Moreover, both strategies complement each other and, when used jointly, result in improved performance compared to when used individually.

4.4. Qualitative Analysis

In Fig. 7, we visualize the learned point sets and the corresponding prediction results of several representative layers (1^{th}, 2^{th}, 6^{th}). It can be observed that learned point sets tend to be located at extreme points of objects progressively. On the easy and medium cases, point sets can easily localize target precisely in the early layers. However, in the hard case, the early diffusion layers (1^{th}, 2^{th}) still generate ambiguous points that may lead to erroneous results. Afterward, the latter layer (6^{th}) refines the ambiguous points to focus on the target. The visualizations verify the effectiveness of point set diffusion and its flexibility in handling complex scenarios.

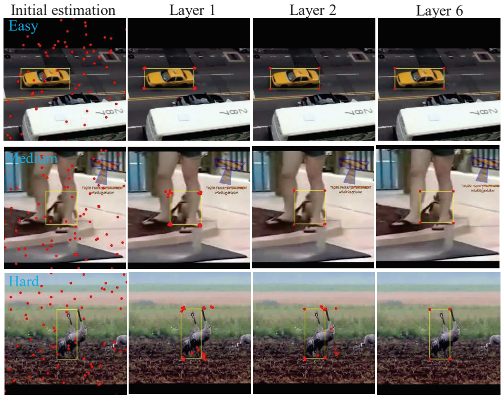


Figure 7. Visualization of some representative diffusion layers. We select easy, medium, and hard cases to see the results of each layer. The target estimations whose confidence scores are under 0.1 are filtered out. Better viewed with zoomed in.

Model Arch.	Threshold	AO(%)	Speed(FPS)
$(E_1 + D_1) \times 1$	0.8	**74.1**	30
$(E_{\frac{1}{3}} + D_{\frac{1}{3}}) \times 3$	0.8	72.3	**45**

Table 3. Early exit in various model architecture. AO Performance is evaluated on GOT-10k [39] benchmark.

4.5. Limitation

The speed-up in Tab. 2e is relatively small. Thus, we construct an interleaved encoder-decoder structure ($(E_{\frac{1}{3}} + D_{\frac{1}{3}}) \times 3$), which stacks 4 encoder layer and 2 decoder layer as repetitive unit. u In Tab. 3, though model (E4+D2)×3 can raise speed notably (30 to 45 FPS), the performance drop is still not satisfying. The architectural dynamics for better trader-off is worth investigating in future work.

5. Conclusion

In this work, we are the first to utilize diffusion-based models for VOT, by reformulating visual tracking as a Noise-to-Target process. DiffusionTrack enjoys several dynamic properties and can gradually localize the target from the background. We further apply point sets into the diffusion process, which can effectively handle complex tracking scenarios. Our DiffusionTrack achieves SOTA results and may attract others to develop a generative tracking paradigm.

Acknowledgments. This work was supported by NSFC (62322113, 62376156), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and the Fundamental Research Funds for the Central Universities.

References

[1] Tomer Amit, Eliya Nachmani, Tal Shaharbany, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021. 2

[2] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *NIPS*, 2021. 2

[3] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022. 2

[4] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *ICLR*, 2022. 2

[5] Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip H S Torr. Fully-convolutional siamese networks for object tracking. In *ECCVW*, 2016. 1

[6] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019. 2, 6

[7] Emmanuel Asiedu Brempong, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. Denoising pretraining for semantic segmentation. In *CVPR*, 2022. 2

[8] Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. Robust object modeling for visual tracking. In *CVPR*, 2023. 6

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[10] 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*, 2022. 6, 7

[11] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *CVPR*, 2023. 1, 2, 4, 5

[12] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022. 2, 5

[13] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. A generalist framework for panoptic segmentation of images and videos. In *ICCV*, 2023. 1, 2

[14] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021. 1, 2, 6

[15] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *CVPR*, 2023. 2, 4, 6

[16] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, 2020. 2, 6

[17] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *CVPR*, 2022. 1, 2, 6

[18] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient convolution operators for tracking. In *CVPR*, 2017. 2

[19] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, 2019. 2, 6

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NIPS*, 2021. 1

[21] 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. 2, 4, 6

[22] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, 2019. 2

[23] 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. 6

[24] Wanshu Fan, Yen-Chun Chen, Dongdong Chen, Yu Cheng, Lu Yuan, and Yu-Chiang Frank Wang. Frido: Feature pyramid diffusion for complex scene image synthesis. *ArXiv*, 2022. 2

[25] Xie Fei, Wang Chunyu, Wang Guangting, Cao Yue, Yang Wankou, and Zeng Wenjun. Correlation-aware deep tracking. In *CVPR*, 2022. 1, 2, 6

[26] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *CVPR*, 2021. 2

[27] Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *CVPR*, 2023. 6

[28] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv*, 2022. 2

[29] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *arXiv*, 2022. 2

[30] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2022. 2

[31] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In *CVPR*, 2020. 1

[32] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. In *NIPS*, 2022. 2

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[34] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 4

[35] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2

[36] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. In *ICVS*, 2008. 2

[37] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NIPS*, 2020. 3

[38] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv*, 2022. 2

[39] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 2018. 2, 6, 7, 8

[40] Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. *arXiv*, 2022. 2

[41] Hyosoon Jang, Sangwoo Mo, and Sungsoo Ahn. Diffusion probabilistic models for graph-structured prediction. *arXiv*, 2023. 2

[42] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. Ddp: Diffusion model for dense visual prediction. In *ICCV*, 2023. 1, 2

[43] Boah Kim, Yujin Oh, and Jong Chul Ye. Diffusion adversarial representation learning for self-supervised vessel segmentation. *arXiv preprint arXiv:2209.14566*, 2022. 2

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

[45] Sungwon Kim, Heeseung Kim, and Sungroh Yoon. Guided-tts 2: A diffusion model for high-quality adaptive text-to-speech with untranscribed data. *arXiv*, 2022. 2

[46] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 2

[47] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 7

[48] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Hyung Jin Chang, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, et al. The tenth visual object tracking vot2022 challenge results. In *ECCV*, 2022. 7

[49] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[50] Alon Levkovitch, Eliya Nachmani, and Lior Wolf. Zero-shot voice conditioning for denoising diffusion tts models. *arXiv*, 2022. 2

[51] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018. 1, 2

[52] 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, 6

[53] Xin Li, Chao Ma, Baoyuan Wu, Zhenyu He, and Ming-Hsuan Yang. Target-aware deep tracking. In *CVPR*, 2019. 2

[54] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In *NIPS*, 2020. 2, 5

[55] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2210.08933*, 2022. 2

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

[57] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 6

[58] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2

[59] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv*, 2017. 6

[60] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *CVPR*, 2015. 2

[61] Fan Ma, Mike Zheng Shou, Linchao Zhu, Haoqi Fan, Yilei Xu, Yi Yang, and Zhicheng Yan. Unified transformer tracker for object tracking. In *CVPR*, 2022. 6

[62] Ziang Ma, Linyuan Wang, Haitao Zhang, Wei Lu, and Jun Yin. Rpt: Learning point set representation for siamese visual tracking. *arXiv*, 2020. 2

[63] K.K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, 2018. 2

[64] Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *ICCV*, 2021. 6

[65] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool. Transforming model prediction for tracking. In *CVPR*, 2022. 2, 6

[66] 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. 6, 7

[67] Hyeonseob Nam and Bohyung Han. Learning multi–domain convolutional neural networks for visual tracking. In *CVPR*, 2016. 2

[68] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021. 5

[69] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *PMLR*, 2022. 2

[70] Jiahao Nie, Zhiwei He, Yuxiang Yang, Mingyu Gao, and Jing Zhang. Glt-t: Global-local transformer voting for 3d single object tracking in point clouds. In *IJCAI*, 2023. 2

[71] Matthieu Paul, Martin Danelljan, Christoph Mayer, and Luc Van Gool. Robust visual tracking by segmentation. In *ECCV*, 2022. 6

[72] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *ICML*, 2021. 2

[73] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv*, 2022. 1, 2

[74] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2015. 2

[75] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 4, 6

[76] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 5

[77] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *ArXiv*, 2022. 2

[78] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv*, 2022. 2

[79] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *TPAMI*, 2022. 1

[80] Jianbing Shen, Yuanpei Liu, Xingping Dong, Xiankai Lu, Fahad Shahbaz Khan, and Steven CH Hoi. Distilled siamese networks for visual tracking. *TPAMI*, 2021. 6

[81] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

[82] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 3

[83] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5

[84] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3

[85] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NIPS*, 32, 2019. 3

[86] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *NIPS*, 2020. 3

[87] Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Transformer tracking with cyclic shifting window attention. In *CVPR*, 2022. 6

[88] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[89] Jaesung Tae, Hyeongju Kim, and Taesu Kim. Editts: Score-based editing for controllable text-to-speech. *arXiv*, 2021. 2

[90] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *CVPR*, 2019. 2

[91] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 4

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

[93] Guangting Wang, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. Spm-tracker: Series-parallel matching for real-time visual object tracking. In *CVPR*, 2019. 2

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

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

[96] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *CVPR*, 2018. 6

[97] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021. 2

[98] 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*, pages 13763–13773, 2021. 6, 7

[99] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. *arXiv*, 2021. 2

[100] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv*, 2021. 2

[101] Qiangqiang Wu, Tianyu Yang, Ziquan Liu, Baoyuan Wu, Ying Shan, and Antoni B. Chan. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In *CVPR*, 2023. 6

[102] Shoule Wu and Ziqiang Shi. Itôtts and itôwave: Linear stochastic differential equation is all you need for audio generation. *arXiv*, 2021. 2

[103] Fei Xie, Lei Chu, Jiahao Li, Yan Lu, and Chao Ma. Videotrack: Learning to track objects via video transformer. In *CVPR*, 2023. 2

[104] Wei Xing, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *CVPR*, 2023. 2

[105] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. SiamFC++: towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, 2020. 1, 2

[106] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, 2021. 1, 2, 6

[107] Bin Yan, Xinyu Zhang, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Alpha-refine: Boosting tracking performance by precise bounding box estimation. *CVPR*, 2021. 2, 7

[108] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV*, 2022. 6

[109] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *arXiv*, 2022. 2

[110] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *arXiv*, 2022. 2

[111] Tianyu Yang and Antoni B Chan. Learning dynamic memory networks for object tracking. In *ECCV*, 2018. 2

[112] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *CVPR*, 2019. 2, 3, 4

[113] Botao Ye, Hong Chang, Bingpeng Ma, and Shiguang Shan. Joint feature learning and relation modeling for tracking: A one-stream framework. *arXiv*, 2022. 1, 2, 4, 6, 7

[114] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *ACM MM*, 2016. 5

[115] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, 2019. 2

[116] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv*, 2022. 2

[117] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, 2019. 1, 2

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

[119] Zhipeng Zhang, Yihao Liu, Xiao Wang, Bing Li, and Weiming Hu. Learn to match: Automatic matching network design for visual tracking. In *ICCV*, 2021. 6

[120] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv*, 2019. 2