MIMTrack: In-Context Tracking via Masked Image Modeling

Xingmei Wang[1], Guohao Nie[1*], Jiaxiang Meng[1*], Zining Yan[2]

[1]College of Computer Science and Technology, Harbin Engineering University
[2]College of Design and Engineering, National University of Singapore
{wangxingmei, nieguhao, mjxwjy}@hrbeu.edu.cn, zn_yan@nus.edu.sg

Abstract

Current Siamese and Transformer trackers commonly use various subtask branches like regression and classification to predict object states. Despite the demonstrated success, these subtask branches might introduce location and scale offsets due to discrepancies and misalignment in the respective predictions. To address this, we propose a novel generative tracker, **MIMTrack**, which defines tracking as a Masked Image Modeling (MIM) process combined with in-context learning (ICL). MIMTrack begins with building the visual prompt image, which consists of a template, a search area, and two target images associated with them. The target image transforms the bounding box into a unified RGB image space as other tracking image. All states prediction are naturally aligned by pixels generation of search target image. In light of this, we perform a MIM process within the visual prompt to reconstruct a masked search target image using the context from other parts. MIM with ICL makes use of implicit cross-relations between template and search area. A single-stream generative framework reduces the offset in the estimation. Furthermore, a latent memory module is introduced as a plugin to enhance pixel generation by leveraging various target appearances over time. The advanced performance observed on leading benchmark datasets highlights the simplicity and effectiveness of our MIMTrack framework.

Introduction

Visual object tracking (VOT) aims to estimate the state of an object in subsequent video frames according to its initial position and scale (Han et al. 2022; Javed et al. 2023). Traditional methods, including Siamese (Li et al. 2019; Bhat et al. 2019) and Transformer (Chen et al. 2021; Ye et al. 2022) trackers, typically formulate VOT as a template-matching task, as shown in Figure 1(a). Multiple subtasks are used to predict various properties of the target—such as location, scale, corner points, and offsets—on a fused search feature (Gao et al. 2022; Cui et al. 2022). Despite their success, these approaches face two significant challenges: (1) Each branch has a distinct goal but shares the same parameters for feature extraction. This can result in difficulties in synchronizinge location and scale estimations during the learning process, potentially affecting the performance of each individual subtask. These optimization discrepancies hinder accurate understanding of tracked objects. (2) Because of the independent operation of multiple branches, varying prediction accuracies can lead to misalignment issues. A predicted box with high classification confidence may still suffer from poor regression accuracy. Although the offset branch has attempted to align the accuracy (Danelljan et al. 2019), the separation of branches still hinder the effective correction of misalignment (Peng et al. 2021).

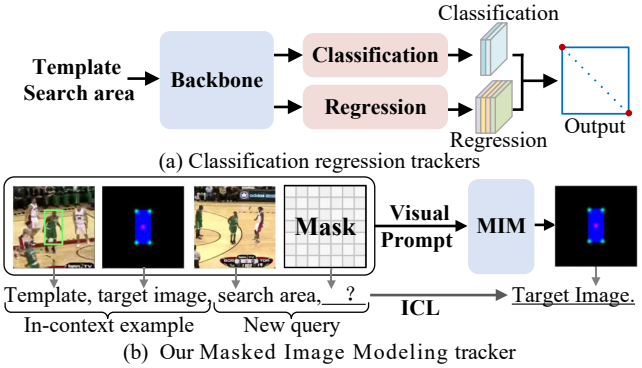


Figure 1: Different pipelines. (a) The bounding box is acquired through classification and regression branches. (b) We treat the template and its target image as an in-context example, and the search area with the masked target image as a new query. Together, they form a visual prompt that allows MIM to predict masked pixels with ICL.

To address these issues, we reformulate VOT as an image generation task based on Mask Image Modeling (MIM) (He et al. 2022). MIM is originally proposed as a visual pre-training technique (He et al. 2022; Xie et al. 2022), involving randomly occluding parts of an image and training the model to reconstruct the missing pixels. Building on this concept, we convert the target box into a target image. Our tracking model uses tracking images, such as templates and search areas, as context to reconstruct the masked pixels of the target image. This pixel-to-pixel approach mimics the ability of human vision to recognize objects from examples. Unlike axis-aligned bounding box-based trackers, our model performs execution of in-context tracking using the

*Corresponding authors.

visual signal as context. By focusing directly on the target area without relying on multi-task learning to decompose the representation, we maintain a consistent output space, which effectively unifies optimization objectives and mitigates misalignment issues. Furthermore, we extend MIM's global context awareness and local feature learning capabilities to tracking, which implicitly models relationships among tracking images and accurately captures the tracking target.

In this work, we present a novel in-context tracking framework using MIM, called **MIMTrack**. MIMTrack transforms the bounding box into an RGB target image to encapsulate the target state and effectively differentiate between foreground and background based on pixel color. In this way, MIMTrack reformulates the tracking as a MIM process combined with in-context learning (ICL). The in-context example consists of a template and its target image to indicate the target's position. The search region is then combined with its masked target image to form a new query. As shown in Figure 1 (b), we integrate the in-context example, and the query into a single visual prompt image. MIMTrack employs an encoder-decoder architecture for the MIM process, where the encoder extracts visual prompt features and the decoder generates the masked pixels in the query. During inference, the bounding box of the target image is reconstructed to provide a general tracking result. Thus, MIMTrack interprets the tracking object within successive frames through direct visual perception. Our unified prediction effectively reduces the gap between model prediction and tracking tasks, addressing potential problems related to multitask divergence and communication issues. In addition, extra examples can deepen the understanding of the context (Bar et al. 2022). To expand the temporal context, a latent memory module generates a few learnable prompt tokens, facilitating memory compression and retrieval. As a result, MIMTrack adapts to target change scenarios and balances efficiency with effectiveness. The key contributions of this paper are as follows:

- We reformulate tracking task as pixel-to-pixel generation. The carefully designed visual prompt unifies the model's input and output spaces.
- We apply ICL withc the MIM-based tracking. The single-stream generative framework aligns target state estimation offsets across the multiple branches.
- To adapt to target changes, we build a latent memory to simply integrate historical appearances in MIM. Extensive experiments demonstrate the effectiveness of MIMTrack, which achieved an AO score of 75.3% on GOT-10k, surpassing SeqTrack, OSTrack, etc.

Related Work

Visual Tracking Framework

Existing tracking datasets (Fan et al. 2019; Huang, Zhao, and Huang 2019) typically use bounding boxes to label objects, indicating their center position, width, and height. This method comprises multiple sub-tasks to ascertain various target state attributes: classification for foreground-background prediction and regression for estimating the target scale. Popular Siamese (Li et al. 2019) and Discriminative trackers (Danelljan et al. 2019; Bhat et al. 2019; Mayer et al. 2022) are primarily built on this architecture. Following this design, transformer trackers extract robust features (Chen et al. 2021; Ye et al. 2022; Lin et al. 2022). Various work (Cai, Liu, and Wang 2024; Xie et al. 2024) have explored spatio-temporal feature modeling. Additionally, approaches like STARK (Yan et al. 2021) and AIATrack (Gao et al. 2022) use separate networks to predict the distribution probabilities of two object corners. Consequently, these isolated sub-tasks require combined losses to train each prediction branch, including crossentropy loss (Chen et al. 2021), focal loss (Ye et al. 2022; Xu et al. 2020; Lin et al. 2017), mean square error loss (Danelljan et al. 2019), IoU loss (Chen et al. 2021; Gao, Zhou, and Zhang 2023), etc. Inspired by Pix2seq (Chen et al. 2022), certain methods (Chen et al. 2023c; Zheng et al. 2023; Wei et al. 2023) transform vision problem into natural language processing (NLP) ones. The bounding box is interpreted as a sequence of language units. An autoregressive head predicts the coordinates symbols sequentially based on the vision feature in order. Nonetheless, the gap of image-to-sequence is addressed through complex designs including causal relation and cross-modal fusion. We attribute the misalignment and optimization divergence problems to the inconsistency of the model output space. Therefore, MIMTrack reformulates the tracking tasks as a pixel-to-pixel generation problem within a unified image space. It overcomes these constraints by generating images to describe the target states. The image-level prediction also avoids quantization errors in the downsampled output.

In-context Learning for Masked Image Modeling

ICL, the latest NLP paradigm, involves completing text from prompts and examples (Brown et al. 2020). It enables models to perform real-time reasoning and recognize new patterns, advancing many NLP tasks. ICL has also demonstrated its effectiveness in computer vision tasks, such as visual-language tasks (Alayrac et al. 2022) and multi-task visual models (Wang et al. 2023c). The in-context examples are used to indicate different visual tasks. In VOT task, visual prompts are also used to integrate multimodal information (Zhu et al. 2023; Luo et al. 2023), thus mitigating the need to retrain tracker in traditional fine-tuning paradigms.

MIM occludes a part of the image and predicts the missing pixels (Zhang et al. 2022; Liao et al. 2023; Chen et al. 2023a). This image inpainting task has been introduced into self-supervised representation learning for downstream tasks following fine-tuning (He et al. 2022; Xie et al. 2022). Subsequent studies have examined the impact of various factors, such as pixel reconstruction (Liu et al. 2023), local multi-scale information (Wang et al. 2023a), in-context example selection (Zhang, Zhou, and Liu 2024) and occlusion regions (Chen et al. 2023b).

Recently, an image prompt-guided MIM model is devised to perform various visual tasks (Bar et al. 2022; Wang et al. 2023b). This prompt combines image examples with a query to generate a resulting image for the masked region based on different task references. In tracking (Zhao, Wang,

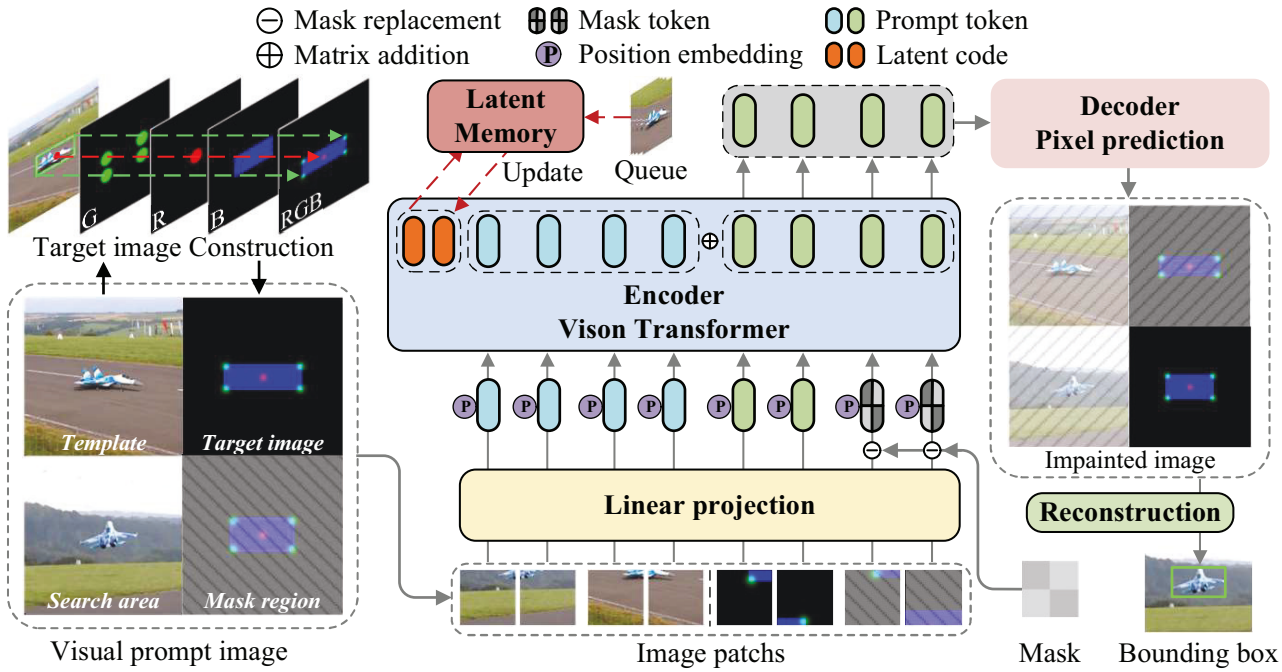


Figure 2: The overview of our MIMTrack framework. The bounding box is mapped to an RGB target image. The search target image is the masked region of the visual prompt image. After linear projection, the mask operation is performed at the token level. The latent memory module introduces previous appearances for feature encoding. The decoder is used to recover the pixels of the mask part, which is reconstructed as the general result.

and Lu 2023), MIM enhances target specific features by reconstructing the template and search area, but isolated feature learning weakens performance. Inspired by the above method, we consider objects as the focus of ICL rather than task types. The tracking task can be described as continuous reasoning given an in-context example. We develop a visual prompt to indicate the tracking object instance. MIM is used to exploit ICL-based tracking. The task is conceptualized as the directional generation of the target image corresponding to the search area. To our knowledge, no prior art has adopted MIM to distinguish tracked objects.

Methodology

Preliminaries

Visual Tracking. Given the bounding box $\mathbf{b_z}$ in the template image $\mathbf{z}$, the objective of visual tracking is to identify the target $\mathbf{b_x}$ within the search area $\mathbf{x}$, as illustrated below:

$$\mathbf{b_x} = f_{\text{VOT}}\left(\mathbf{z}, \mathbf{x}, \mathbf{b_z}\right), \tag{1}$$

where the learned model f_{VOT} can predict the location and scale of the target through multiple branches.

Masked Image Modeling. MIM (Zhang et al. 2022) is a widely used techniqu in vision tasks to enhance feature extraction and improve model robustness. The core idea is to mask certain parts of the image and predict the masked regions based on the context provided by the unmasked regions. Given an original image $\mathbf{I}$ and a binary mask $\mathbf{M}$, the masked image $\mathbf{I}_{\text{mask}}$ is created as $\mathbf{I}_{\text{mask}} = \mathbf{I} \odot \mathbf{M}$. $\odot$ is element-wise multiplication. The prediction can be represented as follows:

$$\mathbf{I}_{\text{pred}} = f_{\text{MIM}}\left(\mathbf{I}_{\text{mask}}\right). \tag{2}$$

The model f_{MIM} predicts the masked pilxes. The objective function is typically used the Mean Squared error (MSE) between $\mathbf{I}$ and $\mathbf{I}_{\text{pred}}$.

Target Image Construction

Traditional tracking methods primarily represent objects using axis-aligned bounding boxes, denoted as $\mathbf{b} = [x, y, w, h]$, which are straightforward to annotate with little ambiguity. However, there exists an inherent inconsistency between the position $[x, y]$ and scale $[w, h]$ estimations. Despite incorporating the offset branch for misalignment (Danelljan et al. 2019), the independent prediction process remains non-communicative. The substantial differences between the input image and the output coordinates lead to complex and divergent tracking predictions. Therefore, we reformulate the pixel representation of the target states to align with the image input space. In unified image space, the joint reasoning of location and scale is able to mitigate optimization divergence and misalignment issues. The consistent spatial relationships among pixels minimize quantization errors.

Taking the original pixels of the target as the output introduces significant ambiguity. The background pixels in the annotation boxes hinder the precise localization of the object. To precisely determine the pixel's position, we solve

this issue as a keypoint prediction problem. For template $\mathbf{z} \in \mathbb{R}^{3\times H\times W}$, $rgb\left(\mathbf{b}_{\mathbf{z}};[H,W]\right)$ is the transition from the bounding box $\mathbf{b}_{\mathbf{z}}$ to the RGB target image $\mathbf{z}_o$. As depicted in the upper left of Figure 2, $\mathbf{z}$ and $\mathbf{z}_o$ have exactly the same dimensions and pixels spatial relationships. The background is represented as black (0 pixel value) in $\mathbf{z}_o$. Then, the corner points are employed to locate the target. In green channel G, a 9×9 Gaussian heatmap marks the target's corner position. The heatmap's peak aligns with the corner's center. Marking all four corners helps mitigate partial occlusion effects.

Nevertheless, it is challenging to estimate the corners that are virtual positions outside the object. We impose constraints on the target image within R and B channels. In the red channel R, the target's center point is marked using a similar Gaussian distribution heatmap to focus on the target itself. In the blue channel B, pixels inside the bounding box are set to 255, and pixels outside are set to 0, effectively directing the model's attention to the coverage area.

MIMTrack framework

Visual prompt image. MIMTrack is a purely vision-based tracking framework in which pixels serve as both model inputs and predicted objects. $\mathbf{x}_o = rgb\left(\mathbf{b}_{\mathbf{x}};[H,W]\right)$ is the target image of the search area $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$. According to the in-context example $[\mathbf{z};\mathbf{z}_o]$, the model is expected to infer approximate representations $\mathbf{x}_o$ for the corresponding search region $\mathbf{x}$. The ICL is incorporated to achieve this pixel-to-pixel target understanding. Inspired by Eq. 2, MIM is employed to model in the unified input-output space:

$$\mathbf{x}_o = f_{\text{MIM}}\left(\mathbf{z};\mathbf{z}_o;\mathbf{x};\mathbf{M}\right). \tag{3}$$

Hence, we synthesize all the tracking images $[\mathbf{z};\mathbf{z}_o;\mathbf{x};\mathbf{x}_o]$ in a single context, namely the visual prompt image $\mathbf{I} \in \mathbb{R}^{2H\times 2W\times C}$. The search target image denotes the masked portion of a visual prompt, with the remaining area serving as the reference in-context. A fixed occlusion strategy has replaced the previously used random mask method (He et al. 2022). As a result, traditional tracking's relation modeling is implicitly embedded in local pixel inpainting. This allows the model to reference both global context and local features for instance-specific information.

Encoder. MIMTrack utilizes a simple encoder-decoder architecture. The encoder employs the ViT (Dosovitskiy et al. 2021) backbone to extract embedding features, utilizing a self-attention mechanism to capture relationships between pixels. MIMTrack uses a simple encoder-decoder architecture. Firstly, the visual prompt image is segmented into patches: $\mathbf{p}_{\text{seg}} \in \mathbb{R}^{3\times N\times S^2}$, where S^2 represents the patch size, and $N = \frac{4HW}{S^2}$ denotes the number of patches. Each image patch $\mathbf{p}_{\text{seg}}$ is transformed into a feature embedding $\mathbf{e} \in \mathbb{R}^{1\times d_m}$ via a linear projection, where d_m denotes the vector dimension. $\mathbf{E} \in \mathbb{R}^{N\times d_m}$ is the embedding features for the entire visual prompt image. Similar to (Xie et al. 2022), we apply a mask operation at token level:

$$\mathrm{E}_{\text{mask}} = \mathrm{E}\odot\mathrm{M} + \mathrm{P}\odot(1-\mathrm{M}), \tag{4}$$

where $\mathbf{M} \in \{0,1\}^N$ is the token-level binary mask, with 0 indicating the masked region. $\mathrm{P} \in \mathbb{R}^{N\times d_m}$ represents a set of placeholders, generated as copies of a learnable placeholder $\mathbf{m} \in \mathbb{R}^{1\times d_m}$. Then, the masked feature of the visual prompt, $\mathbf{E}_{\text{mask}}$, is represented as $[\mathbf{e}_1,\cdots,\mathbf{e}_{3N/4},\mathbf{m},\cdots,\mathbf{m}]$, indicating that the final quarter of tokens (search target image) are replaced by $\mathbf{m}$. Additionally, position embeddings are applied to retain positional information. Finally, $\mathbf{E}_{\text{mask}}$ is fed into the encoder layers for feature updates. To reduce the Transformer computation, the visual prompt is split vertically into two parts for separate feature extraction, and then fused through matrix addition at a specific encoding layer.

Decoder. The decoder remaps the features back to the input resolution. The single-pipeline prediction architecture comprises both simple linear and convolutional layers. We use a linear layer to integrate different depth-encoded features to assist in image restoration. Subsequently, two convolutional layers are employed to predict the masked target image, which is denoted as $x'_o \in \mathbb{R}^{3\times H\times W}$.

Loss function. MIMTrack predicts the raw pixels of the masked region by minimizing the distance between the predicted target image $\mathbf{x}'_o$ and the mapped target image $\mathbf{x}_o$. The loss function $\mathcal{L}$ is employed on the masked pixels:

$$\mathcal{L} = \mathcal{L}_{smooth-\ell 1}\left(\mathbf{x}'_o,\mathbf{x}_o\right) + \mathcal{L}_{ssim}\left(\mathbf{x}'_o,\mathbf{x}_o\right), \tag{5}$$

where $\mathcal{L}_{smooth-\ell 1}$ is the Smooth L1 loss.

Moreover, we aim to achieve precise pixel-level predictions for representing object locations, rather than relying on pre-trained feature representations (Zhang et al. 2022). To supervise local details, we introduce an additional Structural Similarity (SSIM) loss (Wang et al. 2004), denoted as:

$$\mathcal{L}_{ssim} = 1 - \left[l\left(\mathbf{x}'_o,\mathbf{x}_o\right)\cdot c\left(\mathbf{x}'_o,\mathbf{x}_o\right)\cdot s\left(\mathbf{x}'_o,\mathbf{x}_o\right)\right], \tag{6}$$

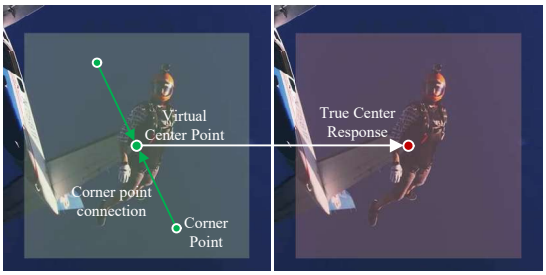
where $\mathcal{L}_{ssim}$ is SSIM loss function. l, c and s mean luminance, contrast and structure metrics, respectively.

Bounding Box Reconstruction

During inference, the unknown search target image is directly specified as the mask region to generate a predicted image $\mathbf{x}'_o$. To evaluate the tracking results, an optimal candidate box is extracted from the predicted target image $\mathbf{x}'_o$. The blue channel represents only a rough extent of the target. For this purpose, the target's center point and the corner points are collectively employed to accurately locate the target, as shown in Figure 3. The local maximum values from the green channel G of image $\mathbf{x}'_o$ are extracted to establish the set of target corner points $\mathbf{C} = [\mathbf{c}_1,\cdots,\mathbf{c}_{N_C}]$. Each corner point $\mathbf{c}_i, (i = 1,\cdots,N_C)$ is characterized by a triple $\left(x_i,y_i,v_i^G\right)$, in which (x_i,y_i) signify the coordinates and v_i^G represents the pixel value. Intuitively, the line of any two corner points can form a bounding box. These candidate bounding boxes can be represented using the adjacency matrix $\mathbf{C}_{\text{adj}}$ of the corner points:

$$\mathbf{C}_{\text{adj}} = \begin{bmatrix} c_{11}^{\text{adj}}\left(\mathbf{c}_1;\mathbf{c}_1\right) & \cdots & c_{1N}^{\text{adj}}\left(\mathbf{c}_1;\mathbf{c}_N\right) \\ \vdots & \ddots & \vdots \\ c_{N1}^{\text{adj}}\left(\mathbf{c}_N;\mathbf{c}_1\right) & \cdots & c_{NN}^{\text{adj}}\left(\mathbf{c}_N;\mathbf{c}_N\right) \end{bmatrix}, \tag{7}$$

where the elements of the matrix $c_{ij}^{\text{adj}}\left(\mathbf{c}_i;\mathbf{c}_j\right)$ represent the summation of two values v_i^G and v_j^G. The score $c_{ij}^{\text{adj}}\left(\mathbf{c}_i;\mathbf{c}_j\right)$



(a) Green channel. (b) Red channel.

Figure 3: Bounding box reconstruction process. (a) The lines connecting the corner points produce a virtual target center position in the green channel. (b) The virtual center position matches a predicted center score in the red channel.

expresses the confidence of the bounding box by the probability of two corner points. The Non-Maximum Suppression is utilized to reduce the number of candidate points, and the self-connections of corner points are eliminated.

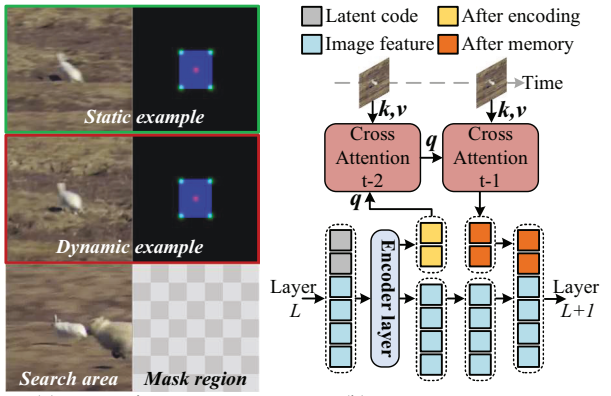
The center of the bounding box naturally coincides with the center of the actual object. We utilize the central response of the R channel to filter the appropriate candidates. For each pair of corners, the midpoint of its diagonal line is extracted to denote a virtual center position of the candidate object. The matrix element $\mathbf{C}_{\text{adj}}$ is updated as follows:

$$\mathbf{C}_{\text{adj}} = v_i^G + v_j^G + \alpha v_{ij}^R \left(\frac{x_i + x_j}{2}, \frac{y_i + y_j}{2} \right), \quad (8)$$

where v_{ij}^R denotes the actual response of the virtual center position in the R channel. α is a control factor between corner points and center point. In the ideal case, the midpoint of the optimal bounding box should coincide exactly with the center of the true target. The maximum value in the adjacency matrix $\mathbf{C}_{\text{adj}}$ represents the desired target. In light of prior experience, a Hann window has been incorporated into the R channel to mitigate the effect of larger displacements between two frames.

Latent memory module

The ensemble of more in-context examples has been shown to provide better predictions (Bar et al. 2022). MIMTrack allows dynamic in-example integration into visual prompts to address appearance changes. The grid arrangement of dynamic prompt image changes from 2×2 to 3×2, as shown in Figure 4(a). With the increase in examples, the dynamic prompts have significantly improved image resolution and tracking performance. However, this strategy is challenging for integrating more appearances and real-time tracking. We propose a latent memory module to replace dynamic example, as shown in Figure 4(b). Some dedicated latent codes $\mathrm{H} \in \mathbb{R}^{N_H \times d_m}$ are added to feature E as compressed internal memory (Burtsev et al. 2020). This extended sequence $[\mathrm{H}; \mathrm{E}]$ is processed using encoder layers, without distinguishing between the latent code and other embedding feature. Then, we separate the memory H_{mem} and input attention flows E in the forward pipeline between encoding layers. A recurrent cross-attention block and a target template



(a) Dynamic prompt. (b) Latent memory.

Figure 4: Updating strategy. (a) Dynamic prompts. (b) Latent memory.

queue are utilized to update the memory flow:

$$\begin{aligned} \mathrm{q} = \mathrm{H}_t^L, \mathrm{k}, \mathrm{v} = \mathrm{E}_{t-2}^{ref}, \\ \mathrm{H}_{t-2}^{L+1} = \ LayerNorm\left(H_t^L + MHA\left(\mathrm{q}, \mathrm{k}, \mathrm{v}\right)\right), \end{aligned} \quad (9)$$

where MHA is Multi-head Attention, and $LayerNorm$ means Layer normalization. H_t^L is the encoder outputs from the previous layer L. The frames E_{t-2}^{ref} of the queue serve as Key k and Value v, while the latent code functions as Query q. Compared to k, v, the number of latent codes q is very limited. Asymmetric attention and cyclical cross-attention block substantially reduce ensemble memory computation. The output latent code H_{t-2}^{L+1} is then passed to the next frame. The final latent code H_{t-1}^{L+1} and $\mathbf{E}$ are reintegrated as $\left[\mathrm{H}_{t-1}^{l+1}; \mathrm{E}\right]$, facilitating information exchange via subsequent encoder layers. The latent code interacts with external memory, while the feature sequence interacts only with the latent code memory. This compels the model to accumulate and reallocate global information through memory.

The dynamic example and reference queue are gradually updated during tracking. To mitigate the update risks, the adjacency score of the tracking results are utilized to select reliable templates. The predicted image with clear outlines and sharp edges indicate a high reliability of the current target, while higher adjacency scores $\mathbf{C}_{\text{adj}}$ tend to be observed. If the adjacency score exceeds the set threshold τ, the current result will be used as a dynamic example or replace the earliest template in the queue.

Experiments

Implementation Details

MIMTrack is trained and tested on a single 3090 GPU using Python 3.8 and PyTorch 1.11.0. The training dataset consists of training splits from COCO (Lin et al. 2014), LaSOT (Fan et al. 2019), GOT-10k (Huang, Zhao, and Huang 2019), and TrackingNet (Muller et al. 2018). Data augmentation techniques involve horizontal flipping and brightness adjustment. The optimizer is AdamW (Loshchilov and Hutter 2017). The learning rate and batch size are set to 1e−4

Methods	GOT-10k			TrackingNet			LaSOT			$\text{LaSOT}_{\text{ext}}$			UAV123
	AO ↑	$\text{SR}_{0.5}$ ↑	$\text{SR}_{0.75}$ ↑	AUC↑	P_{Norm} ↑	P↑	AUC↑	P_{Norm} ↑	P↑	AUC↑	P_{Norm} ↑	P↑	AUC↑
MDNet (Nam and Han 2016)	29.9	30.3	9.9	60.6	70.5	56.5	39.7	46.0	37.3	27.9	34.9	31.8	52.8
SiamRPN++ (Li et al. 2019)	51.7	61.6	32.5	73.3	80.0	69.4	49.6	56.9	49.1	34.0	41.6	39.6	61.3
DiMP (Bhat et al. 2019)	61.1	71.7	49.2	74.0	80.1	68.7	56.9	65.0	56.7	39.2	47.6	45.1	65.4
TrDiMP(Wang et al. 2021)	67.1	77.7	58.3	78.4	83.3	73.1	63.9	-	61.4	-	-	-	67.5
TransT(Chen et al. 2021)	67.1	76.8	60.9	81.4	86.7	80.3	64.9	73.8	69.0	-	-	-	69.1
KeepTrack(Mayer et al. 2021)	-	-	-	-	-	-	67.1	77.2	70.2	48.2	-	-	69.7
STARK(Yan et al. 2021)	68.8	78.1	64.1	82.0	86.9	-	67.1	77.0	-	-	-	-	68.2
AiATrack(Gao et al. 2022)	69.6	63.2	80.0	82.7	87.8	80.4	69.0	79.4	73.8	47.7	55.6	55.4	70.6
SwinTrack-T(Lin et al. 2022)	71.3	81.9	64.5	81.1	-	78.4	67.2	-	70.8	47.6	-	53.9	-
MixFormer-22k(Cui et al. 2022)	70.7	80.0	67.8	83.1	88.1	81.6	69.2	78.7	74.7	-	-	-	70.4
OSTrack(Ye et al. 2022)	71.0	80.4	68.2	83.1	87.8	82.0	69.1	78.7	75.2	47.4	57.3	53.3	70.1
ARTrack(Wei et al. 2023)	73.5	82.2	70.9	84.2	88.7	83.5	70.4	79.5	76.6	-	-	-	67.7
SeqTrack-B (Chen et al. 2023c)	74.7	84.7	71.8	83.3	88.3	82.2	69.9	79.7	76.3	49.5	60.8	56.3	69.2
SeqTrack-L (Chen et al. 2023c)	*74.5*	83.2	*72.0*	**85.0**	89.5	84.9	**72.1**	81.7	79.0	50.5	61.5	*57.2*	-
EVPTrack (Shi et al. 2024)	73.3	83.6	70.7	-	-	-	70.4	80.9	77.2	48.7	59.5	55.1	70.2
STCFormer(Hu et al. 2024)	74.3	*84.2*	**72.6**	-	-	-	71.5	*81.5*	*78.0*	**52.0**	**63.0**	**59.6**	**70.8**
MIMTrack-B	72.6	83.2	69.3	83.1	87.7	80.9	69.1	78.8	75.7	47.7	57.0	54.3	68.2
MIMTrack-B (LM)	73.5	83.8	70.3	*84.1*	*88.8*	*83.9*	70.0	79.9	76.7	48.9	58.2	55.3	69.5
MIMTrack-L	**75.3**	**85.8**	72.5	**85.0**	**89.7**	**85.0**	71.6	**82.0**	**79.7**	*49.6*	*60.3*	57.7	*70.2*

Table 1: Tracking results on four popular benchmarks: GOT-10k, TrackingNet, LaSOT, LaSOText and UAV123. The first three rankings are shown in bold, underline, and italics fonts, respectively.

and 16, respectively. Our model is trained with 500 epochs and 60k matching pairs per epoch. After 400 iterations, the learning rate is reduced by 10 times. ViT-Base (Dosovitskiy et al. 2021) serves as the encoder structure in MIMTrack-B. The decoder receives features from $[5, 8, 11]$ layers of the encoder. The prediction convolutional layer contains 3×3 kernel size, bias weight, activation and normalization. The template and search area are four times the target size and resized to 256^2 pixels. The size of the visual prompt is 512^2 in one example case and 768×512 in two examples. The MIMTrack-L version incorporates a larger backbone network, ViT-Large. We use two in-context examples at both training and testing in MIMTrack-B and MIMTrack-L. The update period and threshold are 4 and 0.7, respectively. For latent memory version, MIMTrack-B (LM), the latent code uses $N_H = 16$ learnable tokens with one in-example. We maintain a template queue of size 128^2 and length 4. The latent code is updated before the feature output layer.

Overall Performance

We assessed our MIMTrack using four well-known large-scale VOT benchmarks and one challenging VOT dataset.

GOT-10k. GOT-10k (Huang, Zhao, and Huang 2019) is a large-scale benchmark with over 10k frames and non-overlapping classes in training and testing to evaluate generalization. According to official protocol, we have trained our MIMTrack only on the GOT-10k training split. Performance metrics include average overlap (AO) and success rate (SR). Table 1 demonstrates that MIMTrack-B surpassed OSTrack (Ye et al. 2022) by 1.6%, 2.8%, 1.1% in AO, $\text{SR}_{0.5}$ and $\text{SR}_{0.75}$. This shows that the MIM method outperforms the multi-task prediction head in similar network settings. With latent memory, the MIMTrack(LM) further improves the AO score to 73.5%. We also obtain significant improvements with an 75.3% AO score using the VIT-Large backbone (MIMTrack-L) . Similar to NLP tasks, in-context tracking also improves context understanding (target) with more parameters. MIMTrack-B achieves a tracking speed of 51.2 FPS, surpassing SeqTrack-B at 50.5 FPS and ARTrack at 20.6 FPS on the same device.

TrackingNet. TrackingNet (Muller et al. 2018) encompasses 511 video sequences depicting various scenes and objects. The area under the curve (AUC), precision (P), and normalized precision (P-Norm) are used to test the tracker. Our MIMTrack-B achieves an impressive 83.1% AUC, which is only slightly lower than SeqTrack by 0.2%, and outperforms SwinTrack by 2% AUC.

LaSOT and LaSOText. The comprehensive long-term tracking datasets, LaSOT (Fan et al. 2019) and LaSOText (Fan et al. 2021), encompass 280 and 150 test video sequences, respectively. Benefiting from latent memory, our MIMTrack-B (LM) effectively resists apparent change. MIMTrack-B (LM) achieves AUC of 70.0% and 48.9% on LaSOT and LaSOText, respectively.

UAV123. UAV123 (Mueller, Smith, and Ghanem 2016), captured by unmanned aerial vehicles, is a video dataset encompassing small-scaled and high-velocity objects. As depicted in Table 1, the proposed MIMTrack-L achieves AUC of 70.2% on UAV123 datasets.

Visualization. We present visual comparisons between 5 advanced trackers and our MIMTrack. As shown in Figure 5, the first and third rows indicate the capability to precisely identify the target's boundaries, while the second and fourth rows exhibit the proficiency in modeling both targets and interfering backgrounds. These findings suggest that MIMTrack excels in consolidating target instance information from visual prompts.

Figure 5: Visualized comparison results.

#	Method	GOT-10k			
		AO↑	$SR_{0.5}$ ↑	$SR_{0.75}$ ↑	FPS↑
1	Baseline	72.6	**83.2**	69.3	51.2
2	1-Conv	71.4	81.4	65.1	53.1
3	2-Linear	71.0	81.2	66.2	53.1
4	8-Vit	**73.0**	83.1	**70.8**	43.8
5	Random	59.8	67.7	50.4	50.5
6	Fusion	64.5	73.9	57.0	52.3
7	Single-layer feature	71.6	81.3	66.7	**55.0**

Table 2: Ablation studies on decoders, mask strategy and feature layers. The optimal results are shown in bold font.

Ablation and Analysis

We use MIMTrack-B without the latent memory module as the baseline model in our ablation study.

Decoders. The rows 2-4 of Table 2 show the effect of the decoder on pixel prediction, including one convolutional layer, two linear layers, and 8 ViT blocks. The 8-ViT decoder shows a slight improvement over the baseline in AO and $SR_{0.5}$. The ViT model performs significantly better on the stricter $SR_{0.75}$ metric but reduces test speed by 7.4 FPS and doubles the training time. In lightweight decoders, a single convolutional layer is slightly less effective than the baseline but still outperforms two linear layers. This may be because local information is more critical than global information for final pixel predictions. We ultimately choose the baseline model for its balance of accuracy and speed.

Mask strategy. We conduct a comparison with the random masking strategy (Xie et al. 2022), which is indicated as *Random* in row 5 of Table 2. At a mask rate equivalent to 25%, the random masking causes a substantial reduction in the AO score by 59.8%. The baseline method, which is able to refer to the target information, maintains stability. In further fusing the two strategies (*Fusion* in row 5 of Table 2), this way achieves an AO score of 64.5%. In general, the model struggles to understand tracking targets when visual prompts are randomly obscured.

#	Number		GOT-10k		
	Train	Test	AO↑	$SR_{0.5}$ ↑	$SR_{0.75}$ ↑
1	1	1	70.8	81.1	66.2
2	2	2	**72.6**	**83.2**	**69.3**
3	2	3	72.1	82.3	67.4

Table 3: Ablation on dynamic visual prompts. The optimal results are shown in bold font.

#	Method	GOT-10k		
		AO↑	$SR_{0.5}$ ↑	$SR_{0.75}$ ↑
1	Baseline	**72.6**	**83.2**	**69.3**
2	SmoothL1 loss	72.2	82.9	69.0
4	L1 loss	72.2	83.1	68.0
5	L2 loss	71.6	81.6	67.7
6	L1+L2 loss	72.1	82.9	67.7

Table 4: Ablation on different loss function. The optimal results are shown in bold font.

Feature layers. We verify the effect of pixel generation by removing two early encoder features, such as the single-layer in Table 2. The absence of fine-grained information has resulted in reductions of approximately 1% in AO, 1.9% in $SR_{0.5}$, and 2.6% in $SR_{0.75}$.

Dynamic visual prompts. For online updating, Table 3 demonstrats distinct scenarios during training and testing phases. With equal training and test examples, the additional in-context example bring a significant performance improvement of 1.6% AO. Limited by device memory, we evaluated the case of more test instances than training instances in *#3*. This creates scenarios where performance is degraded but still better than a single example. It is difficult to shift mode's attention from the exceeded instances to the search area, which may lead to the forgetting phenomenon.

Loss function. To verify the impact of loss on MIM, Table 4 presents the case of individual loss and combined loss. The single SmoothL1 loss achieves the best performance. In baseline, SSIM loss supplement the supervision of local information, which further improves the prediction accuracy.

Conclusions

This paper presents a novel and generative tracker that exploits MIM for in-context tracking. MIMTrack uses the target image to model the target states. Furthermore, we synthesize tracking images as a visual prompt to transfer tracking to a unified image space. The MIM process facilitates the model to carry out tracking using visual signals as contextual information. Benefiting from the latent memory, MIMTrack can easily capture the target change so that the tracker can counteract the appearance change of the target. Quantitative and qualitative analyses conducted in the experiments confirm the positive impact of our approach. In fact, alignment prediction effectively balances traditional localization and classification. This pixel-to-pixel generation facilitates the understanding of the target and eliminates the misalignment caused by multi-task branhes.

Acknowledgments

This work is supported by the Primary Research & Development Plan of Heilongjiang Province (GA23A903); the Key Laboratory of Avionics System Integrated Technology; the Fundamental Research Funds for the Central Universities in China (3072024XX0602).

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.

Bar, A.; Gandelsman, Y.; Darrell, T.; Globerson, A.; and Efros, A. 2022. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35: 25005–25017.

Bhat, G.; Danelljan, M.; Gool, L. V.; and Timofte, R. 2019. Learning Discriminative Model Prediction for Tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Brown, T.; Mann, B.; Ryder, N.; et al. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems: 33*, 1877–1901. Curran Associates, Inc.

Burtsev, M. S.; Kuratov, Y.; Peganov, A.; and Sapunov, G. V. 2020. Memory transformer. *arXiv preprint arXiv:2006.11527*.

Cai, W.; Liu, Q.; and Wang, Y. 2024. HIPTrack: Visual Tracking with Historical Prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19258–19267.

Chen, H.; Zhang, W.; Wang, Y.; and Yang, X. 2023a. Improving Masked Autoencoders by Learning Where to Mask. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, 377–390.

Chen, H.; Zhang, W.; Wang, Y.; and Yang, X. 2023b. Improving masked autoencoders by learning where to mask. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, 377–390. Springer.

Chen, T.; Saxena, S.; Li, L.; Fleet, D. J.; and Hinton, G. 2022. Pix2seq: A Language Modeling Framework for Object Detection. In *International Conference on Learning Representations*.

Chen, X.; Peng, H.; Wang, D.; Lu, H.; and Hu, H. 2023c. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14572–14581.

Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; and Lu, H. 2021. Transformer Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8126–8135.

Cui, Y.; Jiang, C.; Wang, L.; and Wu, G. 2022. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13608–13618.

Danelljan, M.; Bhat, G.; Khan, F. S.; and Felsberg, M. 2019. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4660–4669.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; et al. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint arXiv:2012.00000 [cs].

Fan, H.; Bai, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Harshit; Huang, M.; Liu, J.; et al. 2021. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129: 439–461.

Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; and Ling, H. 2019. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5374–5383.

Gao, S.; Zhou, C.; Ma, C.; Wang, X.; and Yuan, J. 2022. Aiatrack: Attention in attention for transformer visual tracking. In *European Conference on Computer Vision*, 146–164. Springer.

Gao, S.; Zhou, C.; and Zhang, J. 2023. Generalized relation modeling for transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18686–18695.

Han, R.; Feng, W.; Guo, Q.; et al. 2022. Single Object Tracking Research: A Survey. arXiv preprint arXiv:2201.00000 [cs].

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.

Hu, K.; Yang, W.; Huang, W.; Zhou, X.; Cao, M.; Ren, J.; and Tan, H. 2024. Sequential Fusion Based Multi-Granularity Consistency for Space-Time Transformer Tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12519–12527.

Huang, L.; Zhao, X.; and Huang, K. 2019. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE transactions on pattern analysis and machine intelligence*, 43(5): 1562–1577.

Javed, S.; Danelljan, M.; Khan, F. S.; Khan, M. H.; Felsberg, M.; and Matas, J. 2023. 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.

Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; and Yan, J. 2019. SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liao, N.; Shi, B.; Zhang, X.; Cao, M.; Yan, J.; and Tian, Q. 2023. Rethinking visual prompt learning as masked visual token modeling. *arXiv preprint arXiv:2303.04998*.

Lin, L.; Fan, H.; Zhang, Z.; Xu, Y.; and Ling, H. 2022. Swintrack: A simple and strong baseline for transformer tracking. *Advances in Neural Information Processing Systems*, 35: 16743–16754.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference*, 740–755. Springer.

Liu, Y.; Zhang, S.; Chen, J.; Chen, K.; and Lin, D. 2023. PixMIM: Rethinking Pixel Reconstruction in Masked Image Modeling. *Transactions on Machine Learning Research*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Luo, Y.; Guo, X.; Feng, H.; et al. 2023. RGB-T Tracking via Multi-Modal Mutual Prompt Learning. arXiv preprint arXiv:2301.00000 [cs].

Mayer, C.; Danelljan, M.; Bhat, G.; Paul, M.; Paudel, D. P.; Yu, F.; and Van Gool, L. 2022. Transforming Model Prediction for Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8731–8740.

Mayer, C.; Danelljan, M.; Paudel, D. P.; and Van Gool, L. 2021. Learning target candidate association to keep track of what not to track. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13444–13454.

Mueller, M.; Smith, N.; and Ghanem, B. 2016. A benchmark and simulator for uav tracking. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 445–461. Springer.

Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; and Ghanem, B. 2018. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European conference on computer vision (ECCV)*, 300–317.

Nam, H.; and Han, B. 2016. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Peng, J.; Jiang, Z.; Gu, Y.; Wu, Y.; Wang, Y.; Tai, Y.; Wang, C.; and Lin, W. 2021. SiamRCR: Reciprocal Classification and Regression for Visual Object Tracking.

Shi, L.; Zhong, B.; Liang, Q.; Li, N.; Zhang, S.; and Li, X. 2024. Explicit Visual Prompts for Visual Object Tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4838–4846.

Wang, H.; Tang, Y.; Wang, Y.; Guo, J.; Deng, Z.-H.; and Han, K. 2023a. Masked image modeling with local multi-scale reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2122–2131.

Wang, N.; Zhou, W.; Wang, J.; and Li, H. 2021. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1571–1580.

Wang, X.; Wang, W.; Cao, Y.; Shen, C.; and Huang, T. 2023b. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6830–6839.

Wang, X.; Zhang, X.; Cao, Y.; et al. 2023c. SegGPT: Segmenting Everything In Context. arXiv preprint arXiv:2103.16746 [cs].

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Wei, X.; Bai, Y.; Zheng, Y.; Shi, D.; and Gong, Y. 2023. Autoregressive Visual Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9697–9706.

Xie, J.; Zhong, B.; Mo, Z.; Zhang, S.; Shi, L.; Song, S.; and Ji, R. 2024. Autoregressive Queries for Adaptive Tracking with Spatio-Temporal Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19300–19309.

Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; and Hu, H. 2022. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9653–9663.

Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; and Yu, G. 2020. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 12549–12556.

Yan, B.; Peng, H.; Fu, J.; et al. 2021. Learning Spatio-Temporal Transformer for Visual Tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10448–10457.

Ye, B.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2022. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European conference on computer vision*, 341–357. Springer.

Zhang, C.; Zhang, C.; Song, J.; et al. 2022. A Survey on Masked Autoencoder for Self-supervised Learning in Vision and Beyond. arXiv preprint arXiv:2201.00000 [cs].

Zhang, Y.; Zhou, K.; and Liu, Z. 2024. What makes good examples for visual in-context learning? *Advances in Neural Information Processing Systems*, 36.

Zhao, H.; Wang, D.; and Lu, H. 2023. Representation learning for visual object tracking by masked appearance transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18696–18705.

Zheng, Y.; Zhong, B.; Liang, Q.; et al. 2023. Towards Unified Token Learning for Vision-Language Tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023: 1–1.

Zhu, J.; Lai, S.; Chen, X.; Wang, D.; and Lu, H. 2023. Visual prompt multi-modal tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9516–9526.