Integrating Boxes and Masks: A Multi-Object Framework for Unified Visual Tracking and Segmentation

Yuanyou Xu\textsuperscript{1,2}, Zongxin Yang\textsuperscript{1}, Yi Yang\textsuperscript{1,2}  
\textsuperscript{1}ReLER, CCAI, Zhejiang University, China \textsuperscript{2}Baidu Research, China  
\{yoxu, zongxinyang, yangyics\}@zju.edu.cn

Abstract

Tracking any given object(s) spatially and temporally is a common purpose in Visual Object Tracking (VOT) and Video Object Segmentation (VOS). Joint tracking and segmentation have been attempted in some studies but they often lack full compatibility of both box and mask in initialization and prediction, and mainly focus on single-object scenarios. To address these limitations, this paper proposes a Multi-object Mask-box Integrated framework for unified Tracking and Segmentation, dubbed MITS. Firstly, the unified identification module is proposed to support both box and mask reference for initialization, where detailed object information is inferred from boxes or directly retained from masks. Additionally, a novel pinpoint box predictor is proposed for accurate multi-object box prediction, facilitating target-oriented representation learning. All target objects are processed simultaneously from encoding to propagation and decoding, as a unified pipeline for VOT and VOS. Experimental results show MITS achieves state-of-the-art performance on both VOT and VOS benchmarks. Notably, MITS surpasses the best prior VOT competitor by around 6\% on the GOT-10k test set, and significantly improves the performance of box initialization on VOS benchmarks. The code is available at https://github.com/yoxu515/MITS.

1. Introduction

Visual object tracking (VOT) [57, 23, 32, 33] and video object segmentation (VOS) [63, 84, 79, 37] are two critical tasks in computer vision. Visual object tracking involves identifying and tracking specific object(s) in a video stream over time. Video object segmentation aims to segment given object(s) in a video sequence and separate it from the background. Both tasks are essential for applications such as video surveillance and autonomous driving.

\begin{itemize}
\item \textsuperscript{1}Yuanyou Xu worked on this at his Baidu Research internship.
\item \textsuperscript{2}Yi Yang is the corresponding author.
\end{itemize}

In VOT, object sizes and positions are indicated with boxes as box representation, while in VOS object shapes and contours are marked with pixel-level masks as mask representation. Despite their differences, VOT and VOS share similarities. Both tasks require the ability to identify and locate the target object in a video stream accurately in spatial dimension, and to be robust against challenges such as occlusion and fast motion in temporal dimension.

In view of the similarity, tracking and segmentation may have unified multi-knowledge representations [90] and have been explored jointly in some works. 1) Unification. A straightforward solution is to perform conversion between boxes and masks to utilize VOT methods on VOS or VOS methods on VOT. A mask can be converted to a box easily, but generating a mask from a box is hard. Some methods were proposed to address the box-to-mask estimation problem [47, 88, 101]. However, separate but not unified models hinder end-to-end training and are inconvenient to manage in practical applications. 2) Compatibility. Several studies have attempted to unify these two tasks into a single frame-
work. However, some of them [78, 75, 62] still lack compatibility and flexibility in box/mask input/output and resort to extra models. 3) Multi-Object. Despite that some methods [48, 81, 86] possess strong compatibility across VOT and VOS, they mainly focus on the single object scenario and use an ensemble strategy to aggregate the separate result of each object in the multiple object scenario.

Therefore, this paper aims to unif VOS and VOT and improve above shortcomings by integrating boxes and masks in a multi-object framework as Multi-object Integrated Tracking and Segmentation (MITS), as shown in Figure 1. For compatibility problem, a unified identification module is proposed to take both reference boxes in VOT and masks in VOS for initialization. The unified identification module encodes the reference boxes or masks into the unified identification embedding by assigning identities to objects. The coarse identification embedding from boxes is further refined to mitigate the gap between mask and box initialization. The unified identification module is more convenient than borrowing an extra model because it is trained with the whole model in an end-to-end manner.

Besides, the novel pinpoint box predictor is proposed for joint training and prediction with the mask decoder. Previous corner head or center head estimates a box by corners or a center point, which are not have to be inside the object. Emphasizing exterior points may distract learning target-oriented features and affect the mask prediction. To address this problem, we estimate the box by localizing pinpoints, which are always on the edge of the object. However, directly supervise the learning of pinpoints is infeasible due to the lack of annotation. Therefore we perform decoupled aggregation on the pinpoint maps and determine the box only by side-aligned pinpoint coordinates.

All the modules in our framework are not only compatible with two tasks, but also able to process multiple objects simultaneously. The multi-object training and prediction make our framework efficient and robust under complex scenes with multiple objects. Extensive experiments are conducted to demonstrate the strong compatibility and capacity of our framework. Experimental results show that our framework achieves SOTA performance on VOT benchmarks including LaSOT [23], TrackingNet [57] and GOT-10k [32], and VOS benchmark YouTube-VOS [84]. Our method improves 6% over previous SOTA VOT method on GOT-10k, and significantly improves the performance of box initialization on VOS benchmarks. In summary, our contributions are:

• We present a multi-object framework integrating boxes and masks for unified tracking and segmentation.

• The unified identification module is proposed to accept both masks and boxes for initialization.

• A novel pinpoint box predictor is proposed for accurate box prediction together with the mask decoder.

2. Related Work

Visual Object Tracking. In our context, we use visual object tracking (VOT) as a union of single object tracking (SOT) and multi-object VOT, preventing confusion with multi-object tracking (MOT) [56] which mainly considers object association between two detected object sets. VOT has been well studied in recent years. Correlation filter based methods [5, 30, 16, 14, 49, 3, 15, 53] train a correlation filter on training features and perform convolution on test features to predict classification scores for the target object. Siamese approaches [2, 70, 36, 35, 26, 75] use a Siamese network to learn an offline similarity metric between the template and search region, and perform cross-correlation to localize the target object. Recently some works [98, 19, 8, 77, 85, 69] adopt transformers [71] in visual tracking for feature extraction and correlation modeling [83, 13, 97, 7], which achieve promising performance. However, most VOT studies mainly focus on SOT, while ours is the first multi-object VOT framework.

Video Object Segmentation. In video segmentation tasks [39, 41, 38, 12], semi-supervised video object segmentation aims to track given objects with masks rather than boxes in VOT. Online VOS methods [6, 74, 51, 89, 55] fine-tune a segmentation model on the reference mask for each video, while matching-based VOS methods [68, 59, 80, 72, 92, 94] measure the pixel-level similarity to segment target objects. Space-time memory network (STM) [60] leverages a memory network to read object information from past frames with predicted masks, and is further improved and extended by many following works [46, 31, 66, 10, 93, 99]. Since masks are more consuming to annotate than boxes, VOS benchmarks [84, 63] usually have shorter videos than SOT benchmarks [23, 57]. Some methods [40, 42, 9] have been proposed for long-term VOS, but they lack consideration for VOT. AOT [93, 96] and its following works [95, 85, 12] realize simultaneous processing of multiple objects in VOS by the multi-object identification mechanism. We adopt the mechanism in our framework and further improve it for unified tracking and segmentation.

Multi-Object Tracking and Segmentation. MOT [56]/MOTS [73] is quite different from VOT/VOS, and the former relies on object detectors trained on pre-defined object categories, and focuses on the association of detected results. Although some work [81, 86] unified MOT/MOTS and VOT/VOS, we recognize the large gap between these two types of tasks and consider VOT and VOS that can generalize to arbitrary objects as the scope of our work.

Unified Tracking and Segmentation. Prior unified tracking and segmentation methods are listed in Table 1. Wang et al. developed a unifying approach SiamMask

9739
by assigning ID vectors to objects for representing one object. The ID embedding is obtained by encoding masks into an IDentification Embedding by an Information about all the target objects in the reference method AOT [93] and used in its following work [91, 95]. A multi-object identification mechanism is proposed in the VOS framework, and further improve it by integrating both box and mask representation in a Unified IDentification Module, which achieves unified initialization for VOS and VOT.

**Pinpoint Definition & Properties.** Pinpoints are defined as the contiguous points between an object and its axis-aligned rectangle bounding box, which pin the box tightly around the object. Pinpoints are first proposed as Extreme Points for efficient object annotation instead of boxes [61]. We use the name Pinpoints in our paper to emphasize the connection constructed by pinpoints between the box and mask of an object, and further explore its properties for box head design.

**Property 3.1 (Pinpoint Existence)** Given an object and its axis-aligned rectangle bounding box, there is at least one pinpoint on each side of the box. (An example for multiple pinpoints on one box side is in Figure 5.)

**Property 3.2 (Pinpoint-Based Box)** An axis-aligned rectangle bounding box can be fully determined by four parameters, which are the side-aligned coordinates of four pinpoints on four sides respectively, i.e. the y coordinates of top and bottom side pinpoints and the x coordinates of left and right side pinpoints (Illustrated in Figure 4 and 5).

### 3.2. Pipeline Overview

Our unified pipeline for VOT and VOS tasks can be formulated as encoding-propagation-decoding (Figure 2). There are two beliefs for constructing our unified framework. The first is mask-oriented, i.e. extracting and maintaining detailed spatial information as much as possible, which has been proved to be more robust in prior work [48, 88, 62]. The second is multi-object mask-box integrated, i.e. natively supporting simultaneously processing multiple boxes or masks from input to output, which has never been achieved by prior work.

### Unified Box-Mask Encoding.

At the beginning, a reference frame or memory frame $I_m \in R^{H_1 \times W_1}$ is encoded by a backbone encoder (which shares weights for all frames) into the visual embedding $X_m \in R^{HW \times C}$. The reference masks $Y_m \in R^{HW \times N}$ or boxes for N objects are encoded by the Unified Identification Module (Section 3.3) into the Unified IDentification Embedding $E_{id} \in R^{HW \times C}$. Then, the visual embedding is sent to the propagation module to obtain the memory embedding for propagation, together with the ID embedding. Note that any frame with predicted boxes or masks can become a memory frame, and the memory embedding is extended from $R^{HW \times C}$ to $R^{THW \times C}$ with $T$ memory frames.

### Spatial-Temporal Propagation.

For the current frame, the visual embedding obtained from the encoder is fed into the propagation module. In the propagation module, the embedding of current frame is transformed into the query...
Current Image \((t=T)\)

We propose a Unified IDentification Module (UIDM), which has three advantages: 1) It supports both box and mask initialization, keeping rich details in the masks while refining coarse information in the boxes. 2) It encodes the reference in a multi-object way, all objects are integrated in one embedding. 3) It can be directly trained as a part of the whole model in an end-to-end manner.

**Box-Mask Integrated Identification.** The UIDM module consists of two parts, the shared ID bank and the Box ID Refiner (BIDR), as shown in Figure 3. If the reference is mask representation, the shared ID bank generates an ID embedding directly from the masks, which stores rich object information. If the reference is box representation, the boxes are first converted to box-shaped masks, and the shared ID bank generates a coarse ID embedding. Then the coarse ID embedding is refined by BIDR into a fine ID embedding with more object details, which is unified with the mask reference:

\[
E_{id} = \begin{cases} 
ID(Y_m), & \text{if } Y_m \text{ is mask ref.} \\
ID(Y_m) + R(I_m, Y_m), & \text{if } Y_m \text{ is box ref.}
\end{cases}
\]

**Box ID Refinement.** To extract accurate multi-object information from the image according to the boxes, BIDR is designed as a dual path transformer with both self-attention and cross attention layers. One is the image path for global information collection, the other is the object path for local information extraction. First, self-attention is used in both paths. The image path can learn the global information such as occlusion. Then, the dual cross attention is used for

\[
Q_t \in R^{HW \times C} \text{ and the memory embedding is transformed into the key } K_m \in R^{THW \times C} \text{ and value } V_m \in R^{THW \times C}.
\]

The ID embedding \(E_{id} \in R^{THW \times C}\) is fused with \(V_m\) and propagated from memory frames to the current frame via attention mechanism [71, 60, 93, 95]:

\[
\text{Attn}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V, \quad (1)
\]

\[
V_t = \text{Attn}(Q_t, K_m, V_m + E_{id}). \quad (2)
\]

**Dual-Branch Decoding.** Finally, the embedding after propagation is passed to two decoding branches for different prediction. The decoder for predicting masks is a feature pyramid network (FPN) [44]. While the decoder for predicting boxes is a transformer-based pinpoint box head (Section 3.4). Both of these two decoders predict masks or boxes for all objects together. For the mask branch, the predicted logits \(S_m \in R^{H \times W \times M}\) are for object capacity \(M\), and the probability \(P_m \in R^{HW \times N}\) for actual \(N\) objects is obtained by using Softmax on \(N\) channels of \(M\). For the box branch, the predicted probability vectors \(P_b^{x_1, y_2} \in R^{W \times 2M}\) and \(P_b^{y_1, y_2} \in R^{H \times 2M}\) form \(M\) boxes. We choose \(N\) boxes from \(M\) predicted boxes.

**3.3. Unified Identification Module**

For mask-based VOS models, directly taking a box for initialization will lead to severe performance degradation [75, 62], as shown in Table 2. A choice is to borrow an extra model which generates masks from boxes [62]. We argue such a strategy has two shortcomings. First, it is cumbersome: some train another extra model. Second, the extra model can’t be optimized together with the VOS model. To address these problems, we propose a Unified IDentification Module (UIDM), which has three advantages: 1) It supports both box and mask initialization, keeping rich details in the masks while refining coarse information in the boxes. 2) It encodes the reference in a multi-object way, all objects are integrated in one embedding. 3) It can be directly trained as a part of the whole model in an end-to-end manner.
the information exchange between two paths:

\[ V'_O = \text{Attn}(Q_I, K_O, V_O), \quad V'_I = \text{Attn}(Q_O, K_I, V_I) \]  

where the subscript \( I \) is for the image path and \( O \) is for the object path. The local information is integrated into the global image path, which helps to generate finer ID embedding. Meanwhile, the global image information also flows to local objects in the boxes, helping to get better object discriminative information, which is favorable in multi-object scenario. After several transformer layers, the refinement embedding can represent more accurate information about the target objects than boxes. The output of BIDR is added with the coarse ID embedding to get a unified fine ID embedding, and then sent to propagation.

**Auxiliary Mask Reconstruction.** Although the shared ID bank and BIDR in UIDM can be trained in an end-to-end manner, direct learning how to refine a high dimensional embedding may be awkward. We mitigate the problem by employing an extra ID decoder during training. The UIDM acts as an encoder to generate the latent ID embedding, and the ID decoder reconstructs the mask from the latent ID embedding. In this way, the training of UIDM gets easier and more effective. Note that the ID decoder is not used in test.

### 3.4. Pinpoint Box Predictor

Keypoint-based box head is popular in detection and tracking, such as the corner head [34, 87, 13] and the center head [20, 97]. The corner head predicts top left and bottom right points while the center head predicts a box center and box sizes in x and y directions. The corners are exterior points to the objects (even the center does not have to be inside the object). Emphasizing exterior points distracts the learning of accurate object-centric representation, especially when it is shared by both box and mask branch.

To make the box predictor harmonious in representation learning with the mask decoder, we propose a pinpoint box head, which determines a box of an object by localizing its pinpoints, illustrated in Figure 4 and visualized in Figure 5. A prior method [102] has attempted to use pinpoints for object detection, but it still relies on center point for object localization, and requires pinpoint annotations for supervision. However, it is hard to annotate all possible pinpoints for supervision, for example a side-aligned object edge includes infinite pinpoints. To solve this, we propose a decoupled aggregation strategy, which eases the demand for pinpoint annotation for direct supervision.

**Pinpoint Localization.** Given the propagated embedding \( D_c \in R^{HW \times C'} \) as the output from the propagation module, it is fed into several transformer layers with self-attention [71] to perform pinpoint localization. With the powerful global modeling capacity of attention mechanism, object edge features are recognized. Later, pinpoints can be localized by exploring the extreme positions of the edges.

**Decoupled Aggregation.** A box has four or more pinpoints according to Property 3.1, while it can be determined by only four parameters, so it’s redundant to directly predict all the pinpoints to obtain the box. In addition, there is no pinpoint annotation for direct supervision in our training data. Based on Property 3.2, we propose a decoupled aggregation strategy to predict the box by extracting the side-aligned coordinates from pinpoints on four sides. First, we decompose the pinpoints on the top, bottom, left and right sides from the localization feature maps by convolution layers \( W'_{x_l, y_l, z_l, y_l} \in R^{C' \times 4M} \) into four score maps. The score maps are normalized by Softmax into probability maps. Then we horizontally aggregate top/bottom pinpoint probability maps, and vertically aggregate left/right
pinpoint probability maps into probability vectors \( P_b^{x_1,x_2} \in R^{W \times 2^M} \), \( P_b^{y_1,y_2} \in R^{H \times 2^M} \):

\[
P_b^{x_1,x_2} = \sum_y \text{Softmax}(W_{x_1,x_2} \cdot \text{SA}(D_l)), \quad (5)
\]

\[
P_b^{y_1,y_2} = \sum_x \text{Softmax}(W_{y_1,y_2} \cdot \text{SA}(D_l)). \quad (6)
\]

where \( \text{SA} \) is self-attention. Then the boxes for all \( N \) objects are determined by \( [x_1, x_2, y_1, y_1] \in R^{M \times 4} \), which are predicted by \( \text{soft} - \text{argmax}: \)

\[
x_T^1 = C_x \cdot P_b^{x_1}, \quad x_T^2 = C_x \cdot P_b^{x_2} \quad (7)
\]

\[
y_T^1 = C_y \cdot P_b^{y_1}, \quad y_T^2 = C_y \cdot P_b^{y_2} \quad (8)
\]

where \( C_x = [0, 1, ..., W], \quad C_y = [0, 1, ..., H] \) are the coordinate arrangements in x and y directions. Note that we select corresponding \( N \) boxes for \( N \) objects from \( M \).

**Box-Mask Collaboration.** Although a single mask decoder can predict masks and convert them to bounding boxes, there are two advantages of training both a mask decoder and a box predictor: box-mask synergy in training and box-mask consistency during test. We find training two branches together yields better performance than training a single mask branch (Table 5), which proves the pinpoint box branch can promote the learning of better edge-aware representation of objects. In addition, the box representation achieves higher precision than the mask representation on VOT benchmarks. During test, we can also calculate the consistency of the two branches by the box IoU, which can be used to indicate the reliability of the results.

### 3.5. Loss and Optimization

The total loss consists of mask loss and box loss. For mask branch, the loss functions we use are Cross Entropy loss and Jaccard loss, and for box branch we use L1 loss and Generalized IoU loss [64]. For the ID decoder, we only use Cross Entropy loss to supervise the mask reconstruction.

In our framework, both the mask \( Y_0^{\text{mask}} \) and box \( Y_0^{\text{box}} \) reference can be used for initialization. The target is to find parameters \( \theta \) of the model \( f \) for two objectives:

\[
\min_{\theta} L(f(\theta, \ldots, X_{t-1}, X_t, Y_0^{\text{mask}}), Y_t), \quad (9)
\]

\[
\min_{\theta} L(f(\theta, \ldots, X_{t-1}, X_t, Y_0^{\text{box}}), Y_t). \quad (10)
\]

The output of the model \( \hat{Y}_t \) and the ground truth \( Y_t \) include both masks and boxes, but here we simplify them as a single term. The model needs to achieve the minimal total loss both for box and mask initialization. To solve this, we randomly pick one objective and optimize it in each step and alternate between two objectives. The probability of choosing the box format for initialization is set as 0.3.

### 4. Experiment

#### 4.1. Implementation Details

**Model Setting.** The backbone encoder of MITS is ResNet-50 [29]. For transformers in UIDM, propagation module and pinpoint box head, we all set 3 layers. The propagation module can be flexible in our framework, and we use the carefully designed gated propagation module proposed in DeAOT [95] by default. For VOT, we use the box branch for box evaluation by default, and the masks can also be evaluated by taking their bounding boxes.

**Training.** We use the same training data as RTS [62] for fair comparison, including the training sets of LaSOT [23], GOT-10k [32], Youtube-VOS [84] and DAVIS [63]. For VOT datasets, masks are estimated from boxes by STA [101] as in RTS [62]. The backbone encoder, propagation module and mask decoder are pre-trained on static images [22, 45, 11, 67, 27], following prior VOS methods [93, 95], while UIDM and the box predictor are trained from scratch. During training, we use 4 Nvidia RTX3090 GPUs, and the batch size is 16. The model is trained 100,000 steps with an initial learning rate of \( 2 \times 10^{-4} \). The learning rate gradually decays to \( 1 \times 10^{-5} \) in a polynomial manner [92]. We use only one model to evaluate on all benchmarks, except GOT-10k test set, where only the training set of GOT-10k is used for its one-shot evaluation.

#### 4.2. Evaluation on Single Object Tracking

**LaSOT & TrackingNet.** LaSOT [23] is a long-term tracking benchmark consisting of 280 videos in test set. The videos are at 30 FPS and have an average length around 2.5k frames. TrackingNet [57] is a short-term tracking benchmark, which contains 511 videos for test, and the average length is around 0.45k frames. The evaluation metrics are Success Rate (or AUC, Area Under Curve), Precision (P) and Normalized Precision (P_n).

We select very strong SOT methods for comparison in Table 2, such as MixFormer [13] and OSTrack [97], where
### Evaluation on Object Tracking

<table>
<thead>
<tr>
<th>SOT Method</th>
<th>AUC P</th>
<th>AUC S</th>
<th>AO SR (50) SR (70)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiameseNet [57]</td>
<td>33.6 42.0 33.9</td>
<td>57.1 66.3 53.3</td>
<td>34.8 35.3 9.8</td>
</tr>
<tr>
<td>MDMNet [58]</td>
<td>39.7 46.0 37.3</td>
<td>60.6 70.5 56.5</td>
<td>29.9 30.3 9.9</td>
</tr>
<tr>
<td>SiamesePRN++ [35]</td>
<td>49.6 56.9 49.1</td>
<td>73.3 80.0 69.4</td>
<td>51.7 61.6 32.5</td>
</tr>
<tr>
<td>DiMP [3]</td>
<td>56.9 65.0 56.7</td>
<td>74.0 80.1 68.7</td>
<td>61.1 71.7 49.2</td>
</tr>
<tr>
<td>MAMLTrack [76]</td>
<td>52.3 - -</td>
<td>75.7 82.2 72.5</td>
<td>- - -</td>
</tr>
<tr>
<td>Ocean [100]</td>
<td>56.0 65.1 56.6</td>
<td>- - -</td>
<td>61.1 72.1 47.3</td>
</tr>
<tr>
<td>TrDiMP [77]</td>
<td>63.9 - 61.4</td>
<td>78.4 83.3 73.1</td>
<td>67.1 77.7 58.3</td>
</tr>
<tr>
<td>TransT [8]</td>
<td>64.9 73.8 69.0</td>
<td>81.4 86.7 80.3</td>
<td>67.1 76.8 60.9</td>
</tr>
<tr>
<td>STARK [87]</td>
<td>67.1 77.0 -</td>
<td>82.0 86.9 -</td>
<td>68.8 78.1 64.1</td>
</tr>
<tr>
<td>KeepTrack [54]</td>
<td>67.1 77.2 70.2</td>
<td>- - -</td>
<td>- - -</td>
</tr>
<tr>
<td>SBT [83]</td>
<td>66.7 - 71.1</td>
<td>82.2 87.5 -</td>
<td>70.4 80.8 64.7</td>
</tr>
<tr>
<td>ToMP [53]</td>
<td>68.5 79.2 73.5</td>
<td>81.5 86.4 78.9</td>
<td>- - -</td>
</tr>
<tr>
<td>MixFormer [13]</td>
<td>70.1 79.9 76.3</td>
<td>83.9 88.9 83.1</td>
<td>70.7 80.0 67.8</td>
</tr>
<tr>
<td>AiATrack [24]</td>
<td>69.0 79.4 73.8</td>
<td>82.7 87.8 80.4</td>
<td>69.6 80.0 63.2</td>
</tr>
<tr>
<td>OTrack [97]</td>
<td>71.1 81.1 77.6</td>
<td>83.9 88.5 83.2</td>
<td>73.7 83.2 70.8</td>
</tr>
<tr>
<td>SwinTrack [43]</td>
<td>69.6 76.8 74.1</td>
<td>82.5 87.0 80.4</td>
<td>69.4 78.0 64.3</td>
</tr>
</tbody>
</table>

Table 2. Evaluation results on SOT benchmarks. There are two groups of methods 1) SOT only methods, 2) unified tracking and segmentation methods. Special one-shot setting is followed on GOT-10k test. The best two results are shown in red and blue.

### Evaluation on Video Object Segmentation

<table>
<thead>
<tr>
<th>VOS Method</th>
<th>Init</th>
<th>YouTube-VOS 19 Val [84]</th>
<th>DAVIS 17 Val [63]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SiameseMask [78]</td>
<td>B</td>
<td>- - -</td>
<td>- -</td>
</tr>
<tr>
<td>DiMP [3]</td>
<td>- - -</td>
<td>- - -</td>
<td>- -</td>
</tr>
<tr>
<td>SiamesePRN++ [75]</td>
<td>- - -</td>
<td>- - -</td>
<td>- -</td>
</tr>
<tr>
<td>Unicorn [86]</td>
<td>- - -</td>
<td>- - -</td>
<td>- -</td>
</tr>
<tr>
<td>RTS [62]</td>
<td>- - -</td>
<td>- - -</td>
<td>- -</td>
</tr>
</tbody>
</table>

Table 3. Evaluation results on VOS benchmarks. There are two groups of methods 1) VOS only methods, 2) unified tracking and segmentation methods. Different initialization formats, mask (M) or box (B), are compared separately. The best two results are shown in red and blue.

The improvement can be attributed to two aspects. First, UIDM extracts detailed spatial information from boxes, which leads to accurate initialization. Second, MITS propagates the detailed spatial information from multiple memory frames with a long temporal coverage, and the prediction is on the whole image. Considering the videos are at 10FPS, the impressive results suggest that MITS is capable of robustly tracking objects in low frame rate videos, where motion is faster than that in high frame rate videos. SOT methods usually have a strong temporal assumption (searching in a local area) and may fail to capture fast moving objects.

### 4.3. Evaluation on Video Object Segmentation

**YouTube-VOS 2019 & DAVIS 2017.** DAVIS [63] is a small-scale benchmark for VOS, which contains 30 videos with high quality annotation masks in 2017 validation set. YouTube-VOS [84] is a large-scale benchmark for VOS, including 507 videos in 2019 validation set with 65/26 seen/unseen object categories. The Region Jaccard $J$ and the Boundary F measure $F$ are used as metrics.

It can be found in Table 3 that the performance of most prior unified methods is not satisfactory compared with VOS only methods, and MITS achieves SOTA performance on YouTube-VOS. MITS also performs better than all previous unified methods on DAVIS 2017 validation set and also competitive with recent VOS methods [93].
5. Ablation Study

5.1. Unified Identification Module

Ablation results of UIDM are listed in Table 4. First, we compare MITS with DeAOT [95] on the tracking benchmark LaSOT [23]. Since DeAOT is a VOS method, we use off-the-shelf Alpha-Refine [88] model to generate masks from boxes for initialization. Compared with the separate models, our unified framework with UIDM achieves better performance. Second, we provide some ablation results of the box ID refiner (BIDR) in UIDM. During training, masks are reconstructed from the refined ID embedding (Mask Recon.), which leads to 0.5% Success Rate/1.7% Precision improvement. The dual cross attention (Dual CA) is used in BIDR to effectively exchange information between the global image and local cropped boxes. If only a single cross attention for the image path in each layer of BIDR is used, the Success Rate drops 1.1% and Precision drops 2%. The bidirectional information exchange is crucial in extracting detailed spatial information from boxes.

5.2. Pinpoint Box Predictor

Ablation results of the box predictor is in Table 5. We tried corner head, and other variants of the pinpoint head. Compared with a single mask branch, the mask branch is promoted by the box branch during training. Except explicitly predicting the probability maps of pinpoints, another choice is to localize the pinpoints implicitly in high dimension feature maps. The decoupled aggregation on probability maps is replaced by decoupled pooling on localized feature maps. The feature maps are pooled to vectors and then projected to probability vectors. The implicit pinpoint localization performs worse than explicit version on Suc-
In our framework, we employ two branches, a mask decoder and a box predictor for joint training and prediction. Training with two branches yields better results than one branch, while they also have different prediction strategies. Previous work [62] has shown mask representation is more robust than box representation. However, we find the box representation has higher Precision while the mask representation has higher Success Rate, as shown in Table 8. A possible explanation is the different strategies of the two branches for multiple similar objects. If the model fails to tell the objects apart, box predictor often chooses one object but mask decoder often covers multiple possible objects.

Another difference between two branches is on the prediction when the target object is absent. SOT benchmarks like LaSOT [23] ignore the frames when the object is absent (fully occluded or out of view), while VOS benchmarks like YouTube-VOS [84] requires to predict an empty mask with all background. Some SOT methods [13, 97] predicts boxes for all frames without considering whether the object is absent, while VOS methods need to precisely predict a background mask if the object is absent. As a unified framework, MITS can infer whether an object is absent with the help of the mask prediction, as shown in Figure 7.

### 5.3. Propagation Module

The propagation module can be flexible in our framework, and a general form has been shown in Equation 2. In practice, we use the carefully designed Gated Propagation Module (GPM) proposed in DeAOT [95] by default, and the number of GPM layer is 3. We also tried other variants including 2-layer GPM or 3-layer Long Short-Term Transformer (LSTT) proposed in AOT [93]. Readers can refer to AOT [93] and DeAOT [95] for more detailed structures of the propagation modules. As the experimental results show in Table 6, GPM performs better than LSTT on both efficiency and performance. Fewer layers of propagation module and a more lightweight backbone lead to the trade-off between efficiency and performance.

### 5.4. Training Data

There are two options for training data: (A) VOS box&mask + VOT box&pseudo-mask, (B) VOS box&mask + VOT box. (A) is taken as default and we directly borrow the generated pseudo-masks for VOT data from prior work [62]. While due to the strong compatibility of MITS, (B) is also originally supported for training. Table 7 shows that MITS trained with (B) can still achieve strong results. We use different dataset repeats to balance different sources.

### 5.5. Box and Mask Prediction

In our framework, we employ two branches, a mask decoder and a box predictor for joint training and prediction. Training with two branches yields better results than one branch, while they also have different prediction strategies. Previous work [62] has shown mask representation is more
References


Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5374–5383, 2019. 1, 2, 6, 7, 8, 9


[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 7


and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 4, 5, 9


[84] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 1, 2, 6, 7, 9


[95] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. *Advances...


