Video Object of Interest Segmentation

Siyuan Zhou[1]*, Chunru Zhan[2], Biao Wang[2], Tiezheng Ge[2], Yuning Jiang[2], Li Niu[1]†

[1]MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, China
[2]Alibaba Group, Beijing, China
ssluvble@sjtu.edu.cn, {zhanchunru.zcr, eric.wb, tiezheng.gtz, mengzhu.jyn}@alibaba-inc.com, coco235000@gmail.com

Abstract

In this work, we present a new computer vision task named video object of interest segmentation (VOIS). Given a video and a target image of interest, our objective is to simultaneously segment and track all objects in the video that are relevant to the target image. This problem combines the traditional video object segmentation task with an additional image indicating the content that users are concerned with. Since no existing dataset is perfectly suitable for this new task, we specifically construct a large-scale dataset called LiveVideos, which contains 2418 pairs of target images and live videos with instance-level annotations. In addition, we propose a transformer-based method for this task. We revisit Swin Transformer and design a dual-path structure to fuse video and image features. Then, a transformer decoder is employed to generate object proposals for segmentation and tracking from the fused features. Extensive experiments on LiveVideos dataset show the superiority of our proposed method.

1 Introduction

Video object segmentation (VOS) (Perazzi et al. 2016; Pont-Tuset et al. 2017; Xu et al. 2018) refers to the task of segmenting class-agnostic object(s) in a video clip. It has been extensively studied and widely applied in various fields, like augmented reality, autonomous driving, video editing, *etc*. Current researches on VOS has two main paradigms: unsupervised VOS (Song et al. 2018; Wang et al. 2019b; Zhou et al. 2020; Ren et al. 2021) and semi-supervised VOS (Caelles et al. 2017; Perazzi et al. 2017; Voigtlaender et al. 2019a; Lin, Qi, and Jia 2019; Bhat et al. 2020; Liang et al. 2021; Mao et al. 2021; Seong et al. 2021; Xie et al. 2021). The former one aims to automatically segment salient/primary objects while the latter one needs to segment objects specified by either human interaction or initial object annotation in the first frame. In this work, we propose a novel paradigm under the VOS task called Video Object of Interest Segmentation (VOIS). Different from the previous two settings, our new problem aims to simultaneously segment and track all objects in the video that are relevant to a given target image according to the user's interest, as well as requiring no additional annotation during inference. Each target image contains a single target object with white background. A video object is classified as a relevant object only if it looks like the same instance as the given target object in style, pattern, category, and color. Note that a relevant object with geometric deformation in the video is still considered as a relevant object *w.r.t.* the given target object. Figure 1 illustrates a sample video with a target image and ground-truth annotations for the VOIS problem. The new paradigm could facilitate typical applications that require customized choices of objects for segmentation. For example, in advertising live broadcasts, the host may want to highlight the product that he/she is displaying. Under this condition, as long as the host offered a picture of the product in advance, the service provider could use VOIS techniques to obtain the segmentation of relevant products during the live broadcast, and then apply special effects to highlight these identified products in real time.

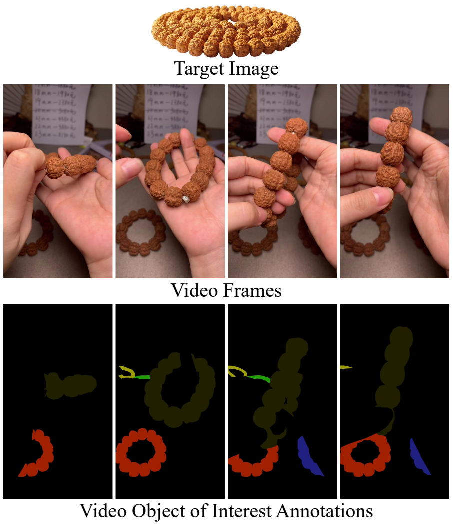


Figure 1: Illustration of video object of interest segmentation (VOIS) with a pair of video clip and target image from LiveVideos dataset. The first row shows the target image. The next two rows display several video frames from the video clip, together with their corresponding VOIS annotations. Masks of the same color across frames belong to the same object.

Video object of interest segmentation is intrinsically a

*Work done during an internship at Alibaba Group.

†Corresponding author.

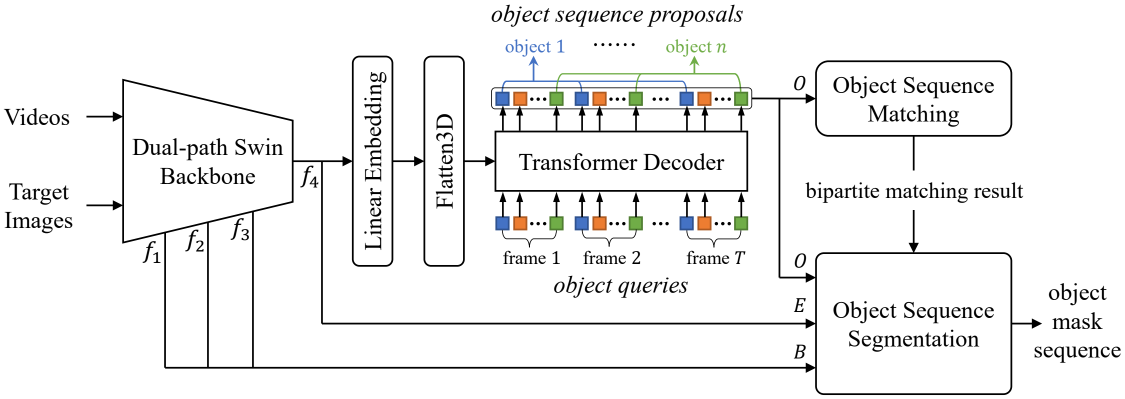


Figure 2: The overall pipeline of our proposed method, which includes a dual-path Swin Transformer Backbone (see Section 5.1), a Transformer Decoder (see Section 5.2), and a object sequence matching/segmentation module (see Section 5.3).

multimodal problem that deals with both a video and an image input. Meanwhile, it requires simultaneous tracking and segmentation of multiple relevant objects in the video. The above two reasons make it more challenging than traditional VOS tasks. VOIS is also related to several existing segmentation tasks. For example, unsupervised video multi-object segmentation (Ventura et al. 2019; Luiten, Zulfikar, and Leibe 2020; Zhou et al. 2020, 2021) aims to segment multiple class-agnostic salient objects in the video, but it can not deal with target objects specified by the user. Video instance segmentation (Yang, Fan, and Xu 2019; Bertasius and Torresani 2020; Wang et al. 2021b; Li et al. 2021) aims to segment class-specific instances in the video, whereas VOIS can deal with objects of unseen categories during inference since the model learns class-agnostic knowledge in training.

To our best knowledge, no previous work deals with video object of interest segmentation, and no existing video dataset is directly applicable to VOIS. Hence, we propose the first large-scale dataset for VOIS, called *LiveVideos*. The new dataset contains 2003 high-resolution live videos and 2418 target images from the E-commerce live broadcast scenes, which constitute 2418 pairs of target image and video for training and inference. Meanwhile, our dataset includes annotations for 3341 video objects and 114k high-quality masks. Our new dataset could serve as a fundamental benchmark for not only video object of interest segmentation, but also traditional video object segmentation. The application scenarios of the dataset includes video retrieval, video highlight, *etc*.

Furthermore, we propose a Transformer-based method for VOIS. As illustrated in Figure 2, the whole framework contains a dual-path Swin Transformer backbone, a Transformer decoder, and a object sequence matching and segmentation module. Swin Transformer (Liu et al. 2021b) has proved to be a well-performed general-purpose network for computer vision. We reuse and reconstruct Swin Transformer to be a dual-path backbone that accepts a 3D video input and a 2D image input simultaneously. Generally, the redesigned Swin Transformer functions as a backbone network that fuses the video feature and the target image feature, and outputs the attended video feature where video regions related to the target image are activated. After that, we introduce a Transformer decoder (Vaswani et al. 2017; Carion et al. 2020) to extract object-level proposals from pixel-level backbone features. Finally, we adopt the instance sequence matching/segmentation module in VisTR (Wang et al. 2021b) to arrange the object proposals according to ground-truth labels, and produce the segmentation masks for each object proposal. Extensive experiments on LiveVideos dataset demonstrate the effectiveness of our method.

In conclusion, the main contributions of this paper are:

- We define and explore a new VOS paradigm called video object of interest segmentation (VOIS).
- We create the first large-scale benchmark for VOIS, containing 2418 pairs of target image and video clip.
- We propose an end-to-end Transformer-based method for VOIS, and prove its advantages over several baselines.

2 Related Work

Video Object Tracking. Video object tracking methods either track objects based on the given bounding boxes in the first frame (*i.e.*, detection-free tracking) (Bertinetto et al. 2016; Nam and Han 2016; Feichtenhofer, Pinz, and Zisserman 2017) or detect and track objects at the same time (*i.e.*, detection-based tracking) (Sadeghian, Alahi, and Savarese 2017; Wojke, Bewley, and Paulus 2017; Son et al. 2017). Both of them only require to produce bounding boxes, whereas video object of interest segmentation requires preciser segmentation masks. Besides, our task aims at objects specified by a target image, which also makes it different from video object tracking. Compared with video multi-object tracking (Voigtlaender et al. 2019b), we have some additional differences: 1) our problem is not limited to moving objects, and 2) if an object goes out of scene for several frames then reappears, the object label should be consistent.

Video Object Segmentation. Video object segmentation (VOS) has two main settings: unsupervised VOS and semi-supervised VOS. The former one (Ventura et al. 2019; Wang et al. 2019a; Lu et al. 2020b; Zhang et al. 2021) segments class-agnostic salient objects, while the latter one (Oh et al. 2018, 2019; Lu et al. 2020a; Park et al. 2021; Duke et al. 2021;

Dataset	Video clips	Categories	Objects	Masks	Exhaustive
FBMS (Ochs, Malik, and Brox 2013)	59	16	139	1.5k	✗
YouTubeObjects (Jain and Grauman 2014)	96	10	96	1.7k	✗
DAVIS2016 (Perazzi et al. 2016)	50	-	50	3.4k	✗
DAVIS2017 (Pont-Tuset et al. 2017)	90	-	205	13.5k	✗
YouTubeVOS (Xu et al. 2018)	4453	94	7755	197k	✗
YouTubeVIS (Yang, Fan, and Xu 2019)	2883	40	4883	131k	✓
LiveVideos	2418	-	3341	114k	✓

Table 1: Statistics of different video segmentation datasets.

Ge, Lu, and Shen 2021; Hu et al. 2021) segments specified objects given in the first frame. Compared with the above two settings, our proposed video object of interest segmentation aims at segmenting video objects relevant to a specified target image. Meanwhile, video object of interest segmentation requires to simultaneously track different relevant objects, whereas traditional VOS mainly focuses on a single object.

Video Instance Segmentation. Video instance segmentation (VIS) (Yang, Fan, and Xu 2019; Athar et al. 2020; Wang et al. 2021b,a; Lin et al. 2021; Liu et al. 2021a) aims to simultaneously detect, track, and segment class-specific instances in a video clip. Compared with VIS, our proposed video object of interest segmentation has two main differences. First, our setting targets at class-agnostic objects, which means it has the potentiality to segment unseen classes during inference. Second, the objects to segment is determined by a specified target image instead of a predefined class set, which makes it more flexible during application because users can arbitrarily decide what type of objects to segment by choosing the target image according to their interest.

3 Video Object of Interest Segmentation

Problem Definition. Given a target image and a video clip with T frames, suppose there are M video objects relevant to the target image. For the i-th object, we use $\mathbf{m}^i_{p...q}$ to denote its binary segmentation mask across the video, where p and $q \in [1, T]$ represents its starting and ending time, respectively. Suppose a video object of interest segmentation algorithm produces H object hypotheses (also called object sequence proposals). For the j-th hypotheses, the algorithm needs to produce a confidence score $s^j \in [0, 1]$ and a sequence of predicted binary masks $\tilde{\mathbf{m}}^j_{\tilde{p}...\tilde{q}}$. The confidence score will be used in the evaluation metric.

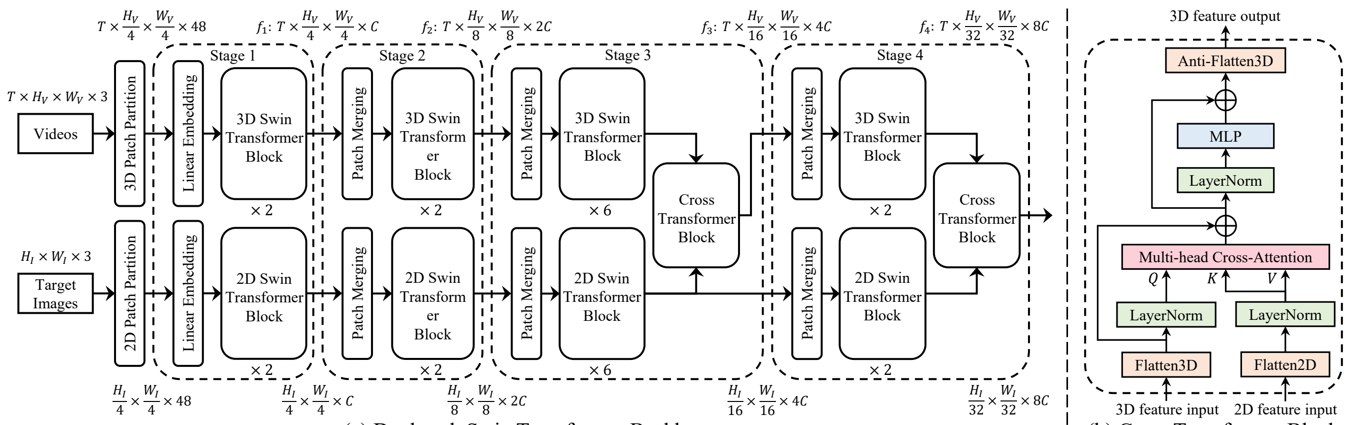
Our objective is to minimize the difference between the hypotheses and the ground truth. It requires that a good VOIS algorithm should be able to 1) correctly detect relevant objects, 2) consistently track all relevant objects across frames, and 3) accurately segment all relevant objects.

Evaluation Metrics. Since VIS (Yang, Fan, and Xu 2019) and VOIS share the same spirit of tracking and segmenting multiple objects simultaneously, we directly adapt the evaluation metrics (average precision AP and average recall AR) in VIS to our VOIS task. AP is defined as the area under the precision-recall curve, and AR is defined as the maximum recall given some fixed number of segmented objects per video. More detailed definitions are given in (Yang, Fan, and Xu 2019). AP and AR work together to reflect the quality of hypotheses produced by the algorithm for evaluation. There is only one difference when we adapt the evaluation metrics to our problem. In VIS, AP and AR are calculated per category and then averaged across the category set. However, in our VOIS problem, since all objects are class-agnostic, AP and AR are directly calculated among all relevant video objects. In other words, evaluation metrics in VOIS can be regarded as a special case of those in VIS, where the category set has only one category, *i.e.*, the relevant object.

4 LiveVideos

Since no existing video segmentation dataset perfectly adapts to our video object of interest segmentation (VOIS) task, we need to establish a new benchmark for this task specifically. There are three important principles to satisfy when we establish the benchmark. The first concern is the source of data. As introduced in Section 1, advertising live broadcasts is a common and practical scenario where VOIS can be applied. This guides us to select data from E-commerce live broadcast scenes to form our dataset. The second concern is the challenge of complex objects like occlusion, appearance change, frequent camera entry/leave, etc. We should take all these conditions into consideration to ensure the diversity and robustness of the dataset. The last concern is the quality of segmentation annotations. We should overcome the weakness of some existing datasets with polygon-based annotations.

Based on the above three principles, we establish a large-scale benchmark called *LiveVideos*. We collect over 10k high-resolution live videos from the E-commerce live broadcast scenes and manually select 2003 representative videos from them. In the broadcast scenes of these selected videos, we collect 2418 relevant target images from the commodity banners, and ensure each of them is clear and not over-covered. Every target image contains a cropped target object with white-base background, indicating the commodity (*e.g.*, clothing, jewellery, daily supplies) displayed in one of the 2003 live videos. Therefore, we have 2418 pairs of live videos and target images. For each pair, we carefully clip out one video clip with a duration of 5.0∼7.2 seconds from the live video, and manually verify that this clip contains the correct target object(s)


$T \times \frac{H_V}{4} \times \frac{W_V}{4} \times 48$
f_1: $T \times \frac{H_V}{4} \times \frac{W_V}{4} \times C$
f_2: $T \times \frac{H_V}{8} \times \frac{W_V}{8} \times 2C$
f_3: $T \times \frac{H_V}{16} \times \frac{W_V}{16} \times 4C$
f_4: $T \times \frac{H_V}{32} \times \frac{W_V}{32} \times 8C$
Stage 1
Stage 2
Stage 3
Stage 4
$T \times H_V \times W_V \times 3$
Videos
3D Patch Partition
Linear Embedding
3D Swin Transformer Block
× 2
Patch Merging
3D Swin Transform er Block
× 2
Patch Merging
3D Swin Transformer Block
× 6
Cross Transformer Block
Patch Merging
3D Swin Transformer Block
× 2
Cross Transformer Block
$H_I \times W_I \times 3$
Target Images
2D Patch Partition
Linear Embedding
2D Swin Transformer Block
× 2
Patch Merging
2D Swin Transform er Block
× 2
Patch Merging
2D Swin Transformer Block
× 6
Patch Merging
2D Swin Transformer Block
× 2
$\frac{H_I}{4} \times \frac{W_I}{4} \times 48$
$\frac{H_I}{4} \times \frac{W_I}{4} \times C$
$\frac{H_I}{8} \times \frac{W_I}{8} \times 2C$
$\frac{H_I}{16} \times \frac{W_I}{16} \times 4C$
$\frac{H_I}{32} \times \frac{W_I}{32} \times 8C$
3D feature output
Anti-Flatten3D
MLP
LayerNorm
Multi-head Cross-Attention
Q
K
V
LayerNorm
LayerNorm
Flatten3D
Flatten2D
3D feature input
2D feature input

(a) Dual-path Swin Transformer Backbone

(b) Cross Transformer Block

Figure 3: (a) The architecture of our dual-path Swin Transformer backbone; (b) Structure of Cross Transformer block. MLP means multi-layer perceptron. Flatten3D means flattening spatial and temporal dimensions. Flatten2D means flattening spatial dimensions. Anti-Flatten3D means recovering the flattened spatial and temporal dimensions.

and is useful for our task (*e.g.*, not too blurry or shaky, no scene transition). After the 2418 video clips are selected, we ask professional human annotators to annotate all the objects (no more than 10 in fact) in each video clip that are relevant to the corresponding target image. We follow (Xu et al. 2018) to adopt a skip-frame annotation strategy. The annotation is performed every four frames in a 20fps frame rate, resulting in a 5fps sampling rate, so no more than 36 frames are annotated in each video clip. Some annotation examples are shown in Figure 1. As a result, our LiveVideos dataset contains 2418 pairs of video clips and target images, and 3341 video objects with 114k high-quality object masks, which form a large-scale benchmark. Table 1 compares LiveVideos with some existing video segmentation datasets. It shows that the scale of our dataset is comparable with YouTubeVOS (Xu et al. 2018) and YouTubeVIS (Yang, Fan, and Xu 2019), and evidently larger than other commonly used datasets.

5 Methodology

Video object of interest segmentation (VOIS) task takes a video clip and a target image as input, and aims to track and segment all video objects that are relevant to the target image. Generally, we tackle the VOIS task with three steps, as shown in Figure 2. First, we design a dual-path Swin Transformer to fuse video features and image features in Section 5.1. Second, we employ a Transformer decoder to generate object proposals from the fused features in Section 5.2. Third, we use a sequence matching module to arrange the object proposals and a sequence segmentation module to produce the segmentation results for each object proposal in Section 5.3.

5.1 Dual-path Swin Transformer

2D Swin Transformer (Liu et al. 2021b) was proposed as a general-purpose backbone to extract image features with a totally end-to-end Transformer-based network. Specifically, it splits the input image into non-overlapping 2D patch tokens and applies four stages to process these tokens. Each stage contains a predefined number of consecutive Swin Transformer Blocks with the proposed 2D window based multi-head self-attention modules (W-MSA) or 2D shifted-window based multi-head self-attention modules (SW-MSA). Four stages work together to produce a hierarchical representation as output. 3D Swin Transformer (Liu et al. 2022) extends the 2D version to deal with video inputs. Likewise, it splits videos into 3D patch tokens, and changes the 2D window based attention modules into 3D versions.

In this work, we combine the 2D version and the 3D version to form a dual-path Swin Transformer that accepts both an image and a video input, as shown in Figure 3. The target image and the video are defined with size $H_\mathrm{I} \times W_\mathrm{I} \times 3$ and $T \times H_\mathrm{V} \times W_\mathrm{V} \times 3$, respectively. We use different subscripts I/V in image/video to avoid confusion. The video has an extra dimension T indicating it contains T frames. We treat each 2D patch of size $4 \times 4 \times 3$ as token in the 2D path, and each 3D patch of size $1 \times 4 \times 4 \times 3$ as token in the 3D path. The 2D patch partitioning layer obtains $\frac{H_\mathrm{I}}{4} \times \frac{W_\mathrm{I}}{4}$ 2D patches and the 3D patch partitioning layer obtains $T \times \frac{H_\mathrm{V}}{4} \times \frac{W_\mathrm{V}}{4}$ 3D patches. Each patch/token consists of a 48-dimensional feature. Then a 2D/3D linear embedding layer is employed to map the 2D/3D token to an arbitrary dimension C.

Like the traditional 2D/3D Swin Transformer, The dual-path Swin Transformer architecture also contains four stages, with each stage combining a 2D stage and 3D stage. Under this design, each stage accepts dual-path inputs and generates dual-path outputs. In the 2D/3D patch merging layer of each stage, the height dimension and the width dimension are down-sampled, while the token dimension doubles. Note that we follow the prior work (Liu et al. 2022) not to down-sample along the temporal dimension in the 3D path. The designs of each 2D/3D Swin Transformer block remain the same as (Liu et al. 2021b, 2022), so we omit the details here.

Since we aim to find video objects of interest conditioned on the target image, we add a Cross Transformer block to fuse video and image feature in stage 3 and stage 4, respectively. The Cross Transformer block contains a multi-head cross-attention and a MLP. We first apply the multi-head cross-attention by treating video feature as query and image feature as key/value. Then, we handle the attended feature via a 2-layer MLP. The output feature of the Cross Transformer block functions as the 3D video input of the next stage. Under this design, video regions relevant to the target image tend to be activated, while the remaining parts tend to be deactivated. Then, the following steps would pay more attention to the video regions that we are interested in, which facilitates segmentation of relevant objects in an implicit way.

For ease of representation, we use f_1, f_2, f_3, and f_4 to represent the output 3D video features of each stage, as shown in both Figure 2 and Figure 3. We finally obtain a 3D video feature f_4 of size $T \times \frac{H_V}{32} \times \frac{W_V}{32} \times 8C$ as the backbone output.

5.2 Transformer Decoder

Motivated by DETR (Carion et al. 2020), we incorporate a Transformer decoder to decode pixel-level features into object-level representations. As illustrated in Figure 2, before the backbone feature f_4 enters the Transformer decoder, we apply a linear embedding layer on f_4 to map it from backbone dimension to decoder hidden dimension. Then, we flatten its spatial and temporal dimensions so that it could be fed into the Transformer decoder. During implementation, we introduce a fixed number of input embeddings to query object-level features, termed as object queries. Specifically, the model decodes n objects for each frame, so the total number of object queries is $N = n \cdot T$. The Transformer decoder works by taking the output of the dual-path Swin Tranformer backbone and N object queries as input to produce N object-level features. After that, N object-level features form n object sequence proposals (*abbr.*, object proposals), and each object proposal is composed by T object-level features from the same index of different frames. The workflow of Transformer decoder is shown in Figure 2. We denote the decoder output by O, which represents the set of n object proposals.

5.3 Object Sequence Matching and Segmentation

Once we obtain object sequence proposals, we will match them with the ground truth object sequences via a object sequence matching module. Then, we predict the mask sequence for each object proposal. After that, we calculate losses between the predicted mask sequence and its matched ground truth sequence to optimize the model, which is achieved by a object sequence segmentation module.

We draw inspirations from VisTR (Wang et al. 2021b) to realize object sequence matching and segmentation. VisTR deals with the video instance segmentation (VIS) task. It first matches its predicted instance sequence proposals with the ground truth sequences via an instance sequence matching module with *bipartite matching loss*. Then, it segments each instance sequence proposal and optimizes the model via an instance sequence segmentation module with a *Hungarian loss*. The above steps of VisTR is somewhat similar to our task, which guides us to adapt the instance sequence matching/segmentation module in VisTR to our required object sequence matching/segmentation module. Generally, there exists two differences between object sequence matching/segmentation and instance sequence matching/segmentation.

The first difference lies in the formation of features. As shown in Figure 2, our object sequence segmentation module follows the instance sequence segmentation module in VisTR to accept three feature inputs: decoder feature O, encoder feature E, and backbone feature B. In our implementation, O remains output feature of the Transformer decoder. $E = f_4$ is exactly the output feature of stage 4 in our dual-path Swin Transformer backbone. $B = \{f_1, f_2, f_3\}$ is the set of multi-level features from the beginning three stages in our dual-path Swin Transformer backbone. The second difference lies in class labels. VIS is a multi-class problem and has a predefined category set. Differently, our VOIS task aims at class-agnostic objects, *i.e.*, single-class objects. Therefore, we modify the category-relevant loss terms in *bipartite matching loss* and *Hungarian loss* from multi-class forms to their single-class counterparts. The *Hungarian loss* contains three parts: classification, box regression, and segmentation. The classification / segmentation part produces the confidence scores / binary masks for object proposals respectively, which are both necessary outputs required by our VOIS task. The detailed loss forms remain unchanged, so we omit here.

6 Experiment

6.1 Dataset and Evaluation Metrics

We conduct experiments on LiveVideos dataset. As introduced in Section 4, we have 2418 pairs of live videos and target images (*i.e.* 2418 samples) to form the whole LiveVideos dataset. We then randomly split the dataset into 1935 training samples and 483 test samples. Each sample is annotated with pixel-level segmentation masks and object labels of all the objects that are relevant to the corresponding target image. We train models on the training set and evaluate them on the test set. The evaluation metrics are Average Precision (AP) and Average Recall (AR) as introduced in Section 3.

6.2 Implementation Details

Network Structure. As introduced in Section 4, each video clip contains no more than 36 frames, so we set the input video sequence length T as 36. The dual-path Swin Transformer backbone is a fusion of 2D Swin Transformer (Liu et al. 2021b), 3D Swin Transformer (Liu et al. 2022) with temporal patch size modified to 1, and two Cross Transformer blocks. The tiny version of 2D/3D Swin Transformer is chosen due to GPU memory limitation. The initial token dimension C is 96, so the backbone output dimension is $8C = 768$. Each Cross Transformer block contains a multi-head attention (Vaswani et al. 2017) and a MLP, with short-cut connections. MLP is composed of two linear layers, with GELU non-linearity in between. The Transformer decoder follows the structure in DETR (Carion et al. 2020), which contains 6 decoder layers with the hidden dimension modified to 384. The Transformer decoder decodes $n = 10$ objects for each frame. The linear embedding layer between the Swin backbone and the Transformer decoder is a linear layer that maps the backbone dimension 768 to the decoder hidden dimension 384. The object sequence matching/segmentation module follows the design in VisTR (Wang et al. 2021b), with the classification head modified from 41 classes (40 YouTube-VIS

Method	Backbone	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack R-CNN (Yang, Fan, and Xu 2019)	ResNet-50 (He et al. 2016)	29.0	46.4	32.1	32.5	35.5
VisTR (Wang et al. 2021b)	ResNet-50 (He et al. 2016)	34.9	54.1	37.0	38.3	41.8
VisTR (Wang et al. 2021b)	ResNet-101 (He et al. 2016)	37.0	56.4	39.2	38.1	43.7
Ours	Dual-path Swin Transformer	38.8	61.6	41.8	41.2	46.6

Table 2: Quantitative comparison between different methods.

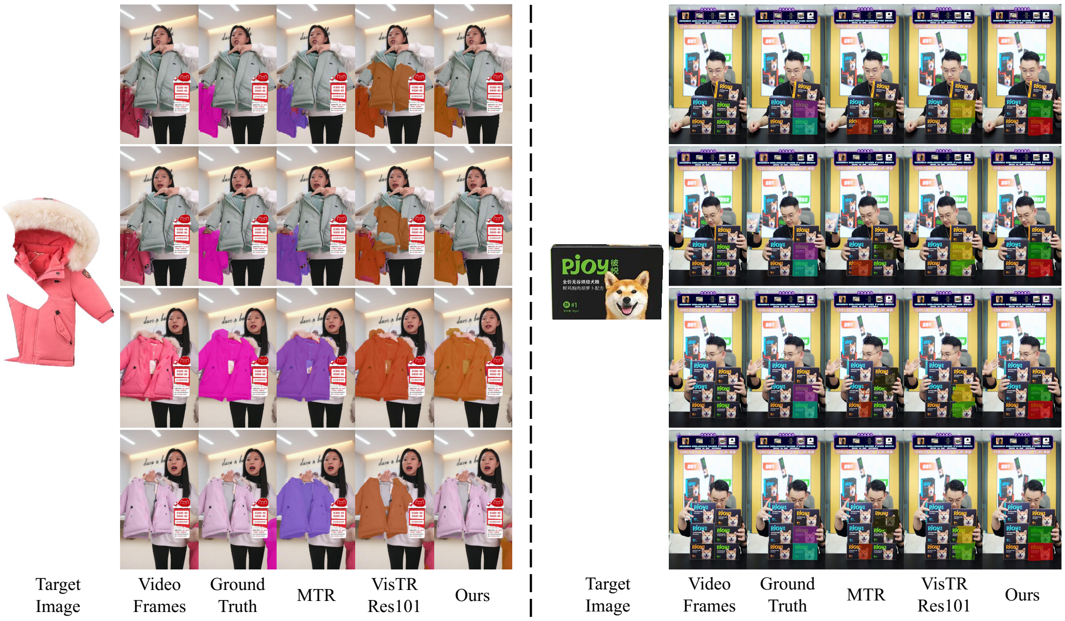


Figure 4: Visualization of video object of interest predictions of different methods. MTR is short for MaskTrack R-CNN. For each method, segmentation masks of the same color across different frames belong to the same object. Zoom in for more details.

categories and background category) to 2 classes (relevant category and background category). The other hyper-parameters in the backbone network (*resp.*, Transformer Decoder, object sequence matching/segmentation) follow the default settings in Swin Transformer (*resp.*, DETR, VisTR).

Data Preprocessing. We first augment the input videos and target images with random horizontal flip and random crop. Then, we resize their shorter edges to 224 by keeping the aspect ratio unchanged. Finally, we apply normalization before feeding them into the network.

Optimization. We adopt AdamW (Loshchilov and Hutter 2017) optimizer with learning rate being 10^{-5} for the dual-path Swin Transformer backbone and 10^{-4} for the remaining parts. The model is trained with 18 epochs, where the learning rate decays by 10x after 12 epochs. We initialize the backbone network with the weights of Swin Transformer pretrained on ImageNet (Deng et al. 2009), and initialize the Transformer decoder with weights of DETR pretrained on MS COCO (Lin et al. 2014). The model is trained on 32 Tesla V100 GPUs with distributed parallel. Each GPU card deals with one pair of video clip and target image in one batch. We perform inference on a single V100 GPU, and retain object proposals with confidence scores larger than 0.001. Experiments are conducted with PyTorch-1.7 (Paszke et al. 2019).

6.3 Baselines

To our best knowledge, no existing method directly adapts to our task. Therefore, we absorb ideas from related tasks to form baselines. Video instance segmentation (VIS) (Yang, Fan, and Xu 2019) is similar to our task in that it also requires to track and segment multiple objects. To adapt a VIS method to a VOIS method, the modification consists of three aspects. First, we should modify the input of the network so that it could accept two inputs, *i.e.*, a video input and an image input. Second, we should fuse video features and image features at a typical stage in the network. Third, we should change the multi-class classification head of the network to its single-class counterpart to deal with class-agnostic output. In this work, we choose to adapt from two representative VIS methods to form our baselines: MaskTrack R-CNN (Yang, Fan, and Xu 2019) and VisTR (Wang et al. 2021b).

MaskTrack R-CNN. MaskTrack R-CNN (Yang, Fan, and Xu 2019) absorbs the 'tracking-by-detection' idea from multi-object tracking to form its method. Typically, it add a tracking head with an external memory into the classical Mask R-CNN to track object instances across frames. To adapt MaskTrack

Target image path	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
	26.7	42.9	28.1	31.0	38.0
✓	38.8	61.6	41.8	41.2	46.6

Table 3: Ablation study on utility of target image path.

Stage 3	Stage 4	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
✓		37.5	60.3	41.0	39.9	44.6
	✓	37.7	59.8	41.4	39.5	45.5
✓	✓	38.8	61.6	41.8	41.2	46.6

Table 4: Ablation study on position of Cross Transformer.

R-CNN (He et al. 2017) to our task, we add a secondary ResNet (He et al. 2016) backbone to accept target image input. Then, we use a Cross Transformer block to fuse video frame features and target image features outputted by two backbones. The fused features are sent to the original network structures after the original video backbone.

VisTR. VisTR (Wang et al. 2021b) is the first Transformer-based VIS method that treats VIS as a direct end-to-end parallel sequence prediction problem. To adapt VisTR to our task, we similarly add a secondary ResNet (He et al. 2016) backbone to accept target image input. Then we add a Cross Transformer block after each Transformer encoder layer to fuse features. The Cross Transformer block takes the output video feature of the current encoder layer and the target image feature from secondary backbone as input to produce a fused feature, which functions as the input video feature of the next encoder layer. Since there six Transformer encoder layers, we have six Cross Transformer blocks correspondingly. The output of the final Cross Transformer block is sent to the instance sequence matching/segmentation module.

6.4 Main Results

Table 2 presents the comparison results between baseline methods and our proposed method. Generally speaking, our proposed method achieves the best results among different methods. In detail, our method surpasses MaskTrack R-CNN by 9.8 AP / 8.7 AR_1, and surpasses VisTR with ResNet-101 backbone by 1.8 AP / 3.1 AR_1. It proves that our method achieves not only better mask quality, but also better temporal consistency and relevant object detection rate. It is noteworthy that although our chosen backbone is merely the tiny version of Swin Transformer, our proposed method still outperforms baselines with ResNet backbones, which implies the great potentiality of our method.

Figure 4 displays two example cases predicted by different methods. As illustrated, our proposed method performs better in the several challenging scenarios: 1) relevant object(s) with heavy motion, 2) relevant object(s) surrounded by multiple confusing objects that are easily misidentified. For the first scenario (left side of Figure 4), our method accurately tracks and segments the relevant object (*i.e.* the pink overcoat) despite its long-distance movement. For the second scenario (right side of Figure 4), our method precisely locates and segments relevant objects with the correct color (*i.e.* boxes with green patterns). Meanwhile, our method seamlessly predicts finer segmentation boundaries. These quantitative results further verify the effectiveness of our proposed method.

6.5 Ablation Studies

Utility of Target Image Path. To prove that our architecture effectively studies the target image information and use it to identify relevant objects in the video clip, we perform an experiment without the target image path. Specifically, the 2D Swin Transformer path and the two Cross Transformer blocks in Figure 3 are deleted. We simply use the 3D Swin Transformer path to extract video frame features, which are then sent to the Transformer decoder. The comparison results are shown in Table 3. Unsurprisingly, the network witnesses a sharp performance drop without the target image input. In detail, AP and AR_1 decrease by 12.1 and 10.2, respectively. The reason for this phenomenon is intuitive. When we do not incorporate target images, the network does not know what object(s) to identify and segment. As a result, it would tend to randomly identify objects in videos during inference, which severally harms its ability to detect user-specified objects.

Position of Cross Transformer Block. In our default implementation, we adopt two Cross Transformer blocks in stage 3 and stage 4, as shown in Figure 3. There are two reasons why we do not place Cross Transformer blocks in the beginning two stages. First, features in the beginning two stages are low-level features in shallow network layers, so they are not expressive enough for the model to find relevant information from the target image. Second, features in the beginning two stages have comparably large spatial dimensions, which makes it space-consuming to compute attention weights in the Cross Transformer block. Considering the above two points, we only use Cross Transformer in the deeper two layers. Table 4 presents a comparison to examine whether each Cross Transformer block in stage 3 / stage 4 is necessary for the overall performance gain. Deleting the Cross Transformer block in stage 3 (*resp.*, stage 4) gives rise to the performance drop of 1.1 AP / 0.7 AR_1 (*resp.*, 1.3 AP / 0.3 AR_1). The comparison shows that both Cross Transformer blocks contribute to the final segmentation results, so we maintain both of them as our default setting.

7 Conclusion

In this work, we present a new task named video object of interest segmentation (VOIS) and specifically construct a large-scale dataset called Livevideos for this task. We also propose an end-to-end Transformer-based method to deal with this multi-modal problem. The proposed method is proved to perform well and surpass several baselines. Compared with the traditional VOS tasks, our proposed VOIS task adopts an additional easily available target image as input to specify what kind of object(s) to track and segment in a video, which makes it user-friendly and conveniently applicable in many situations. We believe this work could attract and promote future research on video object of interest segmentation.

Acknowledgments

The work is supported by the National Science Foundation of China (62076162), the Shanghai Municipal Science and Technology Major/Key Project, China (2021SHZDZX0102, 20511100300), and Alibaba Group through Alibaba Innovation Research Program.

References

Athar, A.; Mahadevan, S.; Osep, A.; Leal-Taixé, L.; and Leibe, B. 2020. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*.

Bertasius, G.; and Torresani, L. 2020. Classifying, segmenting, and tracking object instances in video with mask propagation. In *CVPR*.

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. 2016. Fully-convolutional siamese networks for object tracking. In *ECCV*.

Bhat, G.; Lawin, F. J.; Danelljan, M.; Robinson, A.; Felsberg, M.; Gool, L. V.; and Timofte, R. 2020. Learning what to learn for video object segmentation. In *ECCV*.

Caelles, S.; Maninis, K.-K.; Pont-Tuset, J.; Leal-Taixé, L.; Cremers, D.; and Van Gool, L. 2017. One-shot video object segmentation. In *CVPR*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Duke, B.; Ahmed, A.; Wolf, C.; Aarabi, P.; and Taylor, G. W. 2021. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *CVPR*.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2017. Detect to track and track to detect. In *ICCV*.

Ge, W.; Lu, X.; and Shen, J. 2021. Video object segmentation using global and instance embedding learning. In *CVPR*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *ICCV*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hu, L.; Zhang, P.; Zhang, B.; Pan, P.; Xu, Y.; and Jin, R. 2021. Learning position and target consistency for memory-based video object segmentation. In *CVPR*.

Jain, S. D.; and Grauman, K. 2014. Supervoxel-consistent foreground propagation in video. In *ECCV*.

Li, M.; Li, S.; Li, L.; and Zhang, L. 2021. Spatial feature calibration and temporal fusion for effective one-stage video instance segmentation. In *CVPR*.

Liang, S.; Shen, X.; Huang, J.; and Hua, X.-S. 2021. Video object segmentation with dynamic memory networks and adaptive object alignment. In *ICCV*.

Lin, H.; Qi, X.; and Jia, J. 2019. Agss-vos: Attention guided single-shot video object segmentation. In *ICCV*.

Lin, H.; Wu, R.; Liu, S.; Lu, J.; and Jia, J. 2021. Video instance segmentation with a propose-reduce paradigm. In *ICCV*.

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 *ECCV*.

Liu, D.; Cui, Y.; Tan, W.; and Chen, Y. 2021a. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *CVPR*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*.

Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; and Hu, H. 2022. Video swin transformer. In *CVPR*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lu, X.; Wang, W.; Danelljan, M.; Zhou, T.; Shen, J.; and Gool, L. V. 2020a. Video object segmentation with episodic graph memory networks. In *ECCV*.

Lu, X.; Wang, W.; Shen, J.; Tai, Y.-W.; Crandall, D. J.; and Hoi, S. C. 2020b. Learning video object segmentation from unlabeled videos. In *CVPR*.

Luiten, J.; Zulfikar, I. E.; and Leibe, B. 2020. Unovost: Unsupervised offline video object segmentation and tracking. In *WACV*.

Mao, Y.; Wang, N.; Zhou, W.; and Li, H. 2021. Joint inductive and transductive learning for video object segmentation. In *ICCV*.

Nam, H.; and Han, B. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*.

Ochs, P.; Malik, J.; and Brox, T. 2013. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6): 1187–1200.

Oh, S. W.; Lee, J.-Y.; Sunkavalli, K.; and Kim, S. J. 2018. Fast video object segmentation by reference-guided mask propagation. In *CVPR*.

Oh, S. W.; Lee, J.-Y.; Xu, N.; and Kim, S. J. 2019. Video object segmentation using space-time memory networks. In *ICCV*.

Park, H.; Yoo, J.; Jeong, S.; Venkatesh, G.; and Kwak, N. 2021. Learning dynamic network using a reuse gate function in semi-supervised video object segmentation. In *CVPR*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Perazzi, F.; Khoreva, A.; Benenson, R.; Schiele, B.; and Sorkine-Hornung, A. 2017. Learning video object segmentation from static images. In *CVPR*.

Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Van Gool, L.; Gross, M.; and Sorkine-Hornung, A. 2016. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*.

Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; and Van Gool, L. 2017. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*.

Ren, S.; Liu, W.; Liu, Y.; Chen, H.; Han, G.; and He, S. 2021. Reciprocal transformations for unsupervised video object segmentation. In *CVPR*.

Sadeghian, A.; Alahi, A.; and Savarese, S. 2017. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *ICCV*.

Seong, H.; Oh, S. W.; Lee, J.-Y.; Lee, S.; Lee, S.; and Kim, E. 2021. Hierarchical memory matching network for video object segmentation. In *ICCV*.

Son, J.; Baek, M.; Cho, M.; and Han, B. 2017. Multi-object tracking with quadruplet convolutional neural networks. In *CVPR*.

Song, H.; Wang, W.; Zhao, S.; Shen, J.; and Lam, K.-M. 2018. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ventura, C.; Bellver, M.; Girbau, A.; Salvador, A.; Marques, F.; and Giro-i Nieto, X. 2019. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*.

Voigtlaender, P.; Chai, Y.; Schroff, F.; Adam, H.; Leibe, B.; and Chen, L.-C. 2019a. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*.

Voigtlaender, P.; Krause, M.; Osep, A.; Luiten, J.; Sekar, B. B. G.; Geiger, A.; and Leibe, B. 2019b. Mots: Multi-object tracking and segmentation. In *CVPR*.

Wang, T.; Xu, N.; Chen, K.; and Lin, W. 2021a. End-to-end video instance segmentation via spatial-temporal graph neural networks. In *ICCV*.

Wang, W.; Lu, X.; Shen, J.; Crandall, D. J.; and Shao, L. 2019a. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*.

Wang, W.; Song, H.; Zhao, S.; Shen, J.; Zhao, S.; Hoi, S. C.; and Ling, H. 2019b. Learning unsupervised video object segmentation through visual attention. In *CVPR*.

Wang, Y.; Xu, Z.; Wang, X.; Shen, C.; Cheng, B.; Shen, H.; and Xia, H. 2021b. End-to-end video instance segmentation with transformers. In *CVPR*.

Wojke, N.; Bewley, A.; and Paulus, D. 2017. Simple online and realtime tracking with a deep association metric. In *ICIP*.

Xie, H.; Yao, H.; Zhou, S.; Zhang, S.; and Sun, W. 2021. Efficient regional memory network for video object segmentation. In *CVPR*.

Xu, N.; Yang, L.; Fan, Y.; Yue, D.; Liang, Y.; Yang, J.; and Huang, T. 2018. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*.

Yang, L.; Fan, Y.; and Xu, N. 2019. Video instance segmentation. In *ICCV*.

Zhang, K.; Zhao, Z.; Liu, D.; Liu, Q.; and Liu, B. 2021. Deep transport network for unsupervised video object segmentation. In *ICCV*.

Zhou, T.; Li, J.; Li, X.; and Shao, L. 2021. Target-aware object discovery and association for unsupervised video multi-object segmentation. In *CVPR*.

Zhou, T.; Li, J.; Wang, S.; Tao, R.; and Shen, J. 2020. Matnet: Motion-attentive transition network for zero-shot video object segmentation. *IEEE Transactions on Image Processing*, 29: 8326–8338.