TokenCut: Segmenting Objects in Images and Videos with Self-supervised Transformer and Normalized Cut

Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, Dominique Vaufreydaz

TokenCut: Segmenting Objects in Images and Videos with Self-supervised Transformer and Normalized Cut

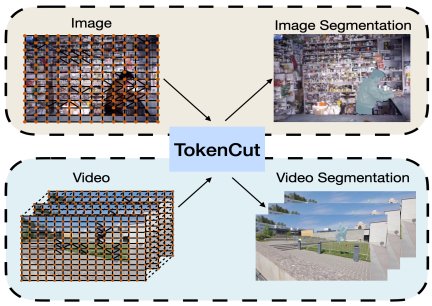
Author Version

Yangtao Wang[1*], **Xi Shen**[2*], **Yuan Yuan**[3], **Yuming Du**[4], **Maomao Li**[2], **Shell Xu Hu**[5]
James L. Crowley[1], **Dominique Vaufreydaz**[1]
[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
[2] Tencent AI Lab [3] MIT CSAIL [4] LIGM (UMR 8049) - Ecole des Ponts, UPE
[5] Samsung AI Center, Cambridge

Abstract

In this paper, we describe a graph-based algorithm that uses the features obtained by a self-supervised transformer to detect and segment salient objects in images and videos. With this approach, the image patches that compose an image or video are organised into a fully connected graph, in which the edge between each pair of patches is labeled with a similarity score based on the features learned by the transformer. Detection and segmentation of salient objects can then be formulated as a graph-cut problem and solved using the classical Normalized Cut algorithm. Despite the simplicity of this approach, it achieves state-of-the-art results on several common image and video detection and segmentation tasks. For unsupervised object discovery, this approach outperforms the competing approaches by a margin of 6.1%, 5.7%, and 2.6% when tested with the VOC07, VOC12, and COCO20K datasets. For the unsupervised saliency detection task in images, this method improves the score for Intersection over Union (IoU) by 4.4%, 5.6% and 5.2%. When tested with the ECSSD, DUTS, and DUT-OMRON datasets. This method also achieves competitive results for unsupervised video object segmentation tasks with the DAVIS, SegTV2, and FBMS datasets. Our implementation is available at https://www.m-psi.fr/Papers/TokenCut2022/.

(a) Attention maps associated to different patches



(b) A unified method for image and video segmentation.

Figure 1: Attention maps associated with different patches highlight different regions of the object (Fig. 1a), which motivates us to build a unified graph-based solution for unsupervised image and video segmentation (Fig. 1b).

1 Introduction

Detecting and segmenting salient objects in an image or video are fundamental problems in computer vision with applications in real-world vision systems for robotics, autonomous driving, traffic monitoring, manufacturing, and embodied artificial intelligence [18, 68, 69]. However, current approaches rely on supervised learning requiring large data sets of high-quality, annotated training data [34]. The high cost of this approach becomes even more apparent when using transfer learning to adapt a pre-trained object detector to a new application domain. Researchers have attempted to overcome this barrier using active learning [1, 48], semi-supervised learning [8, 36], and weakly-supervised learning [26, 43] with limited results. In this paper, we report on results of an effort to use features provided by transformers trained with self-supervised learning, obviating the need for expensive annotated training data.

*Corresponding Author

Vision transformers trained with self-supervised learning [6,14], such as DINO [6] and MAE [17,19,55] have been shown to outperform supervised training on downstream tasks. In particular, the attention maps associated with patches typically contain meaningful semantic information (Fig. 1a). For example, experiments with DINO [6] indicate that the attention maps of the class token highlight salient object regions. However, such attention maps are noisy and cannot be directly used to detect or segment objects.

The authors of LOST [49] have shown that the learned features from DINO can be used to build a graph and segment objects using the inverse degrees of nodes. Specifically, LOST employs a heuristic seed expansion strategy to accommodate noise and detect a single bounding box for a foreground object. We have investigated whether such learned features can be used with a graph-based approach to detect and segment salient objects in images and videos (Fig. 1b), formulating the segmentation problem using the classic normalised cut algorithm (Ncut) [45].

In this paper we describe TokenCut, a unified graph-based approach for image and video segmentation using features provided by self-supervised learning. The processing pipeline for this approach, illustrated in Fig. 2, is composed of three steps: *1)* graph construction, *2)* graph cut, *3)* edge refinement. In the graph construction step, the algorithm uses image patches as nodes and uses features provided by self-supervised learning to describe the similarity between pairs of nodes. For images, edges are labeled with a score for the similarity of patches based on learned features for RGB appearance. For videos, edge labels combine similarities of learned features for RGB appearance and optical flow.

To cut the graph, we rely on the classic normalized cut (Ncut) algorithm to group self-similar regions and delimit the salient objects. We solve the graph-cut problem using spectral clustering with generalized eigen-decomposition. The second smallest eigenvector provides a cutting solution indicating the likelihood that a token belongs to a foreground object, which allows us to design a simple post-processing step to obtain a foreground mask. We also show that standard algorithmns for edge-aware refinement, such as Conditional Random Field [29] (CRF) and Bilateral Solver [5] (BS) can be used to refine the masks for detailed object boundary detection. This approach can be considered as a run-time adaptation method, because the model can be used to process an image or video without the need to retrain the model.

Despite its simplicity, TokenCut significantly improves unsupervised saliency detection in images. Specifically, it achieves 77.7%, 62.8%. 61.9% mIoU on the ECSSD [46], DUTS [65] and DUT-OMRON [73] respectively, and outperforms the previous state-of-the-art by a margin of 4.4%, 5.6% and 5.2%. For unsupervised video segmentation, TokenCut achieves competitive results on DAVIS [40], FBMS [39], SegTV2 [31]. Additionally, TokenCut also obtains important improvement on unsupervised object discovery. For example, TokenCut outperforms DSS [37], which is a concurrent work, by a margin of 6.1%, 5.7%, and 2.6% respectively on the VOC07 [15], VOC12 [16], COCO20K [34].

In summary, the main contributions of this paper are as follows:

- We describe TokenCut , a simple and unified approach to segment objects in images and videos that does not require human annotations for training. *
- We show that TokenCut significantly outperforms previous state-of-the-art methods unsupervised saliency detection and unsupervised object discovery on images. As a training-free method, TokenCut achieves competitive performance on unsupervised video segmentation compared to the state-of-the-art methods.
- We provide a detailed analysis on the TokenCut to validate the design of the proposed approach.

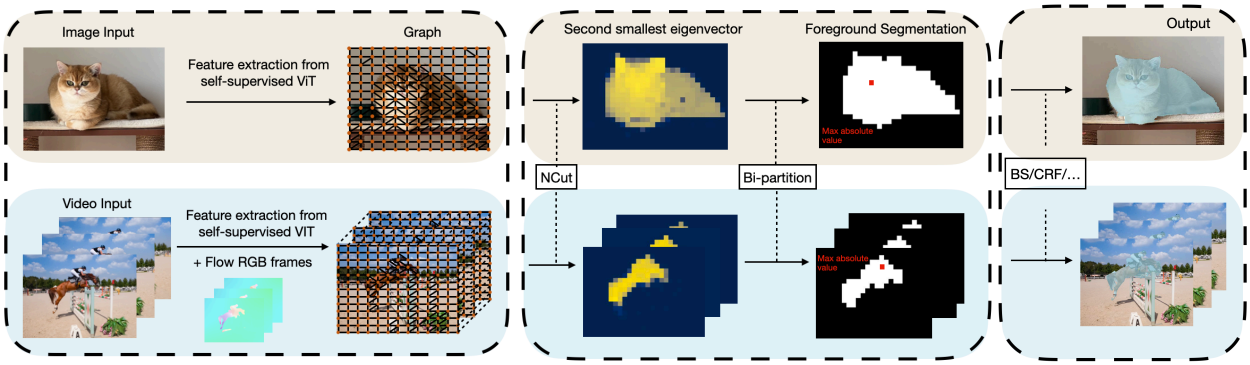
2 Related Work

Self-supervised vision transformers. ViT [14] has shown that the transformer architecture [59] can be effective for computer vision tasks using supervised learning. Recently, many variants of ViT have been proposed to learn image encoders in a self-supervised manner. MoCo-V3 [7] demonstrates that using contrastive learning on ViT can achieve strong results. DINO [6] shows that transformers can be trained with self-distillation loss [20] and shows that the features learn by ViT contain explicit information useful for image semantic segmentation. Inspired by BERT, several approaches [4, 13, 19, 33] learn by missing token replacement, masking some tokens from the input input and learning to recover the missing tokens in the output.

Unsupervised object discovery. Given a group of images, unsupervised object discovery seeks to discover and delimit similar objects that appear in multiple images. Early research [9,21,24,25,60] formulated the problem using an hypothesis about the frequency of object occurrences. Other researchers formulated object detection as an optimization problem over bounding box proposals [11,53,61,62] or as a ranking problem [63].

Recently, LOST [49] significantly improved the state-of-the-art for unsupervised object discovery. LOST extracts features using a self-supervised transformer based on DINO [6] and designs a heuristic seed expansion strategy to obtain a single object region. A concurrent work DSS [37] designs a weighted graph over patches using self-supervised transformer features as well as a KNN based image matting algorithm using a color affinity matrix. The eigen-decomposition of the affinity matrix is computed to obtain a coarse object mask. Following

*The implementation is available at https://www.m-psi.fr/Papers/TokenCut2022/. An online demo is accessible at https://huggingface.co/spaces/yangtaowang/TokenCut (last access May 2023).



(a) Graph Construction (b) Graph Cut (c) Edge Refinement

Figure 2: An overview of the TokenCut approach. The algorithm constructs a fully connected graph in which the nodes are image patches and the edges are similarities between the image patches using transformer features. Object segmentation is then solved using the Ncut algorithm [45]. Bi-partition of the graph using the second smallest eigenvector allows to detect foreground object. A Bilateral Solver [5] (BS) or Conditional Random Field [29] (CRF) can be used for edge refinement.

LOST [49], DSS also assumes that the foreground object occupies a smaller region than the background. While DSS has a similar eigen-attention map as TokenCut, DSS is less able to detect large objects.

As with LOST, TokenCut also uses features obtained with self-supervised learning. However, rather than relying on the attention map of some specific nodes, TokenCut forms a fully connected graph of image tokens, with edges labeled with a similarity score between tokens based on transformer features. The classical Ncut [45] algorithm is then used to detect and segment image objects.

Unsupervised saliency detection. Unsupervised saliency detection seeks to segment a salient object within an image. In this paper, we show that incorporating a simple post-processing step into TokenCut for unsupervised object discovery can provide a strong baseline method for unsupervised saliency detection. Earlier works on Unsupervised saliency detection [23, 32, 71, 79] use techniques such as color contrast [10], background priors [67], or super-pixels [32, 73]. More recently, unsupervised deep models [38, 77] have been used for saliency detection using noisy pseudo-labels generated from different handcrafted saliency methods. [64] shows that unsupervised GANs can differentiate between foreground and background pixels and generate high-quality saliency masks. SelfMask [47] use a spectral clustering method with self-supervised features to group pixels into a set of candidate clusters. SelfMask is trained by selecting the salient masks from the set of spectral clusters as pseudo-masks for supervision using cluster voting scheme.

Unsupervised video segmentation. Given an unlabeled video, unsupervised video segmentation aims to generate pixel-level masks for the object of interest in the video. Prior works segment objects by selecting super-pixels [28], learning flattened 3D object representations [30], constructing an adversarial network to mask a region such that the model can predict the optical flow of the masked region [75], or reconstructing the optical flow in a self-supervised manner [72], etc. DyStaB [74] first partitions the motion field by minimizing the temporal consistent mutual information and then uses the segments to learn the object detector, in which the models are jointly trained with a bootstrapping strategy. The deformable sprites method (DeSprite) [76] trains a video auto-encoder to segment the object of interest by decomposing the video into layers of persistent motion groups. In contrast to these methods [72, 74, 75], our proposed method does not require prior training on videos. Compared with methods [28, 30] that do not train on videos, our method achieves superior performance.

3 Approach: TokenCut

In this section, we present TokenCut, a unified algorithm that can be used to segment salient objects in an image or moving objects in a video. Our approach, illustrated in Fig. 2, is based on a graph where the nodes are visual patches from either an image or a sequence of frames, and the edges are similarities between the features of the nodes based on the features provided by a visual transformer trained with self-supervised learning.

This section is organised as follows: we first briefly review vision transformers and the Normalized Cut algorithm in Section 3.1.1 and Section 3.1.2. We then describe the TokenCut algorithm for object detection and segmentation in images and videos in Section 3.2.

3.1 Background

3.1.1 Vision Transformers

The Vision Transformer has been proposed in [14]. The key idea is to process an image with transformer [59] ar-

chitectures using non-overlapping patches as tokens. For an image with size $H \times W$, a vision transformer takes non-overlapping $K \times K$ image patches as inputs, resulting in $N = HW/K^2$ patches. Each patch is used as a token, described by a vector of numerical features that provide an embedding. An extra learnable token, denoted as a class token CLS, is used to represent the aggregated information of the entire set of patches. A positional encoding is added to CLS token and the set of patch tokens, and the resulting vector is fed to a standard Vision Transformer with self-attention [59] and layer normalization [2].

The Vision Transformer is composed of several stacked layers of encoders, each with feed-forward networks and multiple attention heads for self-attention, paralleled with skip connections. For the TokenCut algorithm, we use the Vision Transformer, trained with self-supervised learning. We extract latent features from the final layer as the input features for TokenCut.

3.1.2 Normalized Cut (Ncut)

Graph partitioning. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of nodes and edges respectively. $\mathbf{E}$ is the similarity matrix with $\mathbf{E}_{i,j}$ as the edge between the i-node v_i and the j-th node v_j. Ncut [45] is proposed to partition the graph into two disjoint sets $\mathcal{A}$ and $\mathcal{B}$. Different to standard graph cut, Ncut criterion considers both the total dissimilarity between $\mathcal{A}$ and $\mathcal{B}$ as well as the total similarity within $\mathcal{A}$ and $\mathcal{B}$. Precisely, we seek to minimize the Ncut energy [45]:

$$\frac{C(\mathcal{A},\mathcal{B})}{C(\mathcal{A},\mathcal{V})} + \frac{C(\mathcal{A},\mathcal{B})}{C(\mathcal{B},\mathcal{V})}, \tag{1}$$

where C measures the degree of similarity between two sets. , $C(\mathcal{A},\mathcal{B}) = \sum_{\mathbf{v}_i \in \mathcal{A}, \mathbf{v}_j \in \mathcal{B}} \mathbf{E}_{i,j}$ and $C(\mathcal{A},\mathcal{V})$ is the total connection from nodes in $\mathcal{A}$ to all the nodes in the graph.

As shown by [45], the equivalent form of optimization problem in Eqn 1 can be expressed as:

$$\min_{\mathbf{x}} E(\mathbf{x}) = \min_{\mathbf{y}} \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{E})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}}, \tag{2}$$

with the condition of $\mathbf{y} \in \{1, -b\}^N$, where b satisfies $\mathbf{y}^T\mathbf{D}\mathbf{1} = 0$, where $\mathbf{D}$ is a diagonal matrix with $\mathbf{d}_i = \sum_j \mathbf{E}_{i,j}$ on its diagonal.

Ncut solution with the relaxed constraint. Taking $\mathbf{z} = \mathbf{D}^{\frac{1}{2}}\mathbf{y}$, Eqn 2 can be rewritten as:

$$\min_{\mathbf{z}} \frac{\mathbf{z}^T\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{E})\mathbf{D}^{-\frac{1}{2}}\mathbf{z}}{\mathbf{z}^T\mathbf{z}}. \tag{3}$$

Indicating in [45], the formulation in Eqn 3 is equivalent to the Rayleigh quotient [58], which is equivalent to solve $\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{E})\mathbf{D}^{-\frac{1}{2}}\mathbf{z} = \lambda\mathbf{z}$, where $\mathbf{D}-\mathbf{E}$ is the Laplacian matrix and known to be positive semidefinite [41]. Therefore $\mathbf{z}_0 = \mathbf{D}^{\frac{1}{2}}\mathbf{1}$ is an eigenvector associated to the smallest eigenvalue $\lambda = 0$. According to the Rayleigh quotient [58], the second smallest eigenvector $\mathbf{z}_1$ is perpendicular to the smallest one ($\mathbf{z}_0$) and can be used to minimize the energy in Eqn 3,

$$\mathbf{z}_1 = \operatorname*{argmin}_{\mathbf{z}^T\mathbf{z}_0} \frac{\mathbf{z}^T\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{E})\mathbf{D}^{-\frac{1}{2}}\mathbf{z}}{\mathbf{z}^T\mathbf{z}}.$$

Taking $\mathbf{z} = \mathbf{D}^{\frac{1}{2}}\mathbf{y}$,

$$\mathbf{y}_1 = \operatorname*{argmin}_{\mathbf{y}^T\mathbf{D}\mathbf{1}=0} \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{E})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}}.$$

Thus, the second smallest eigenvector of the generalized eigensystem $(\mathbf{D}-\mathbf{E})\mathbf{y} = \lambda\mathbf{D}\mathbf{y}$ provides a solution to the Ncut [45] problem.

3.2 The TokenCut Algorithm

The TokenCut algorithm consists of three steps: (a) Graph Construction, (b) Graph Cut, (c) Edge Refinement. An overview of the algorithm is shown in Fig. 2.

3.2.1 Graph construction

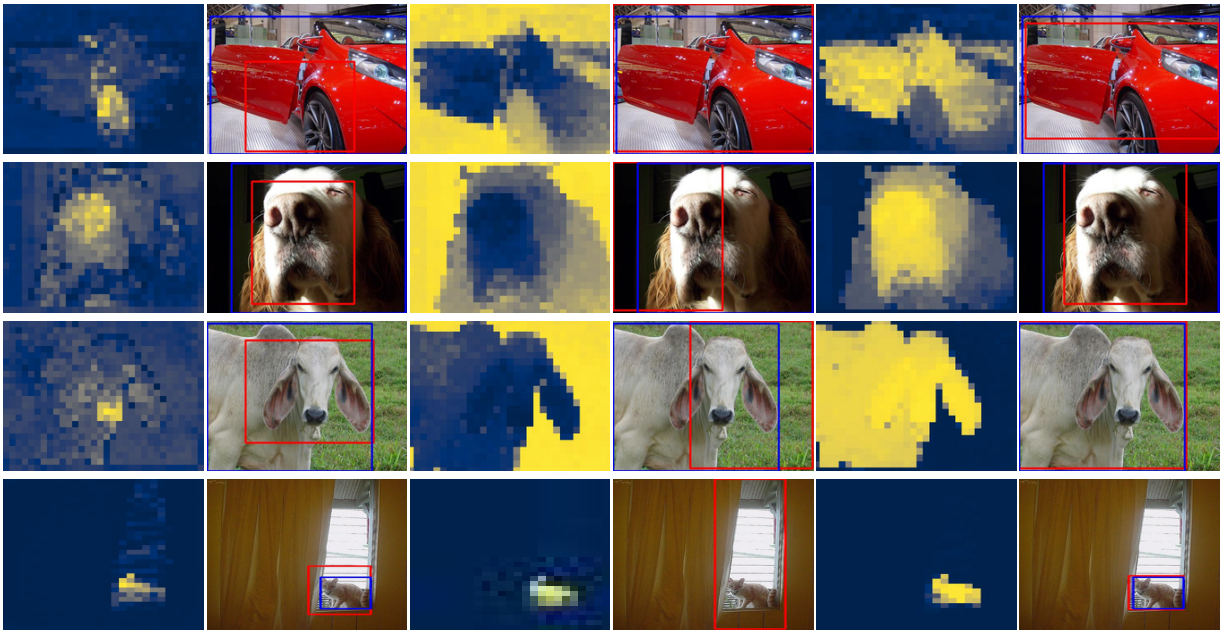
Image Graph. As described in Section 3.1.2, TokenCut operates on a fully connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathbf{v}_i$ represents the feature vectors of the node v_i. Each patch is linked to other patches by labeled edges, $\mathcal{E}$. Edge labels represent a similarity score S.

$$\mathcal{E}_{i,j} = \begin{cases} 1, & \text{if } S(\mathbf{v}_i, \mathbf{v}_j) \geq \tau \\ \epsilon, & \text{else} \end{cases}, \tag{4}$$

where τ is a hyper-parameter and $S(\mathbf{v}_i, \mathbf{v}_j) = \frac{\mathbf{v}_i\mathbf{v}_j}{\|\mathbf{v}_i\|_2\|\mathbf{v}_j\|_2}$ is the cosine similarity between features. ϵ is a small value 10^{-5} to assure a fully connected graph. Note that the spatial location information has been implicitly included in the features, which is achieved by positional encoding in the transformer.

Video Graph. As with images, videos are presented as a fully connected graph where the nodes $\mathcal{V}$ are visual patches and the edges $\mathcal{E}$ are labeled with the similarity between patches. However, for videos, similarity includes a score based on both RGB appearance and a RGB representation of optical flow computed between consecutive frames [3]. The algorithm extracts a sequence of feature vectors using a vision transformer as described in Section 3.1.1. Let $\mathbf{v}_i^I$ and $\mathbf{v}_i^F$ denote the feature of i-th image patch and flow patch respectively. Edges are labeled with the average over the similarities between image feature and flow features, expressed as:

$$\mathcal{E}_{i,j} = \begin{cases} 1, & \text{if } \frac{S(\mathbf{v}_i^I, \mathbf{v}_j^I) + S(\mathbf{v}_i^F, \mathbf{v}_j^F)}{2} \geq \tau \\ \epsilon, & \text{else} \end{cases}. \tag{5}$$



(a) LOST Inverse Degree Attention | (b) LOST Detection | (c) DSS Eigen Attention | (d) DSS Detection | (e) Our Eigen Attention | (f) Our Detection

Figure 3: **Visual results of unsupervised single object discovery on VOC12.** In (a), we show the map of LOST [49] inverse degrees, which is used for detection (b). In (c), DSS [37] eigen attention map is shown for its detection in (d), note that DSS is hard to detect large objects leading to an inverse foreground and background in eigen attention map. For our approach, we illustrate the eigenvector in (e) and our detection in (f). Blue and Red bounding boxes indicate the ground-truth and the predicted bounding boxes respectively.

Image feature provide segmentation using appearance while flow features focus on segmentation with motion. We provide a full analysis on the definition of edges in Section 4.5.

3.2.2 Graph Cut

The Ncut algorithm is used to partition the fully connected graph. Ncut computes the second smallest eigenvector of the generalized eigensystem, as described in Section 3.1.2 to highlight salient objects. We refer to this eigenvector as a measure for "eigen-attention", and provide visualizations of the attention map provided by this vector in Section 4. TokenCut uses eigen-attention to bi-partition the graph, determines which partition belongs to the foreground and then determines the nodes that belong to the each object region.

Bi-partition the graph. To partition the nodes into two disjoint sets, TokenCut uses the average value of the second smallest eigenvector to cut the graph $\overline{\mathbf{y}_1} = \frac{1}{N}\sum_i \mathbf{y}_1^i$. Formally, $\mathcal{A} = \{\mathbf{v}_i | \mathbf{y}_1^i \leq \overline{\mathbf{y}_1}\}$ and $\mathcal{B} = \{\mathbf{v}_i | \mathbf{y}_1^i > \overline{\mathbf{y}_1}\}$. Note that, we also explored the use of classical clustering algorithms, such as K-means and EM, to cluster the second smallest eigenvector into 2 partitions. The comparison is available in Section 4.5. Our experimens show that the average value generally provides better results.

Foreground Determination. Given the two disjoint sets of nodes, TokenCut selects the partition with the maximum absolute value $\mathbf{v}_{max}$ as the foreground. Intuitively, the foreground object should be salient and thus less connected to the entire graph. In other words, $\mathbf{d}_i < \mathbf{d}_j$ if $\mathbf{v}_i$ belongs to the foreground while $\mathbf{v}_j$ is the background token. Therefore, the eigenvector of the foreground object should have a larger absolute value than the background region.

Select the object. In images, we are interested in segmenting a single object. However, the foreground can contain more than one salient object region. TokenCut selects the connected component in the foreground containing the maximum absolute value $\mathbf{v}_{max}$ as the detected object. In videos, as the goal is to segment objects based on both motion and appearance, TokenCut takes the entire foreground region as the final output.

3.2.3 Edge Refinement

The graph cut algorithm provides coarse masks of object regions due to the large size of transformer patches. The boundaries of such masks can be easily refined using standard edge refinement technique. We have experimented with off-the-shelf edge-aware post-processing techniques such as Bilateral Solver [5] (BS), Conditional Random Field [29] (CRF) on top of the obtained coarse mask to

Table 1: **Comparisons for unsupervised single object discovery**. We compare TokenCut to state-of-the-art object discovery methods on VOC07 [15], VOC12 [16] and COCO20K [34, 62] datasets. Model performances are evaluated with CorLoc metric. "Inter-image Simi." means the model leverages information from the entire dataset and explores inter-image similarities to localize objects.

Method	Inter-image Simi.	DINO [6] Feat.	VOC07 [15]	VOC12 [16]	COCO20K [34, 62]
Selective Search [49, 57]		-	18.8	20.9	16.0
EdgeBoxes [49, 80]		-	31.1	31.6	28.8
Kim et al. [27, 49]	✓	-	43.9	46.4	35.1
Zhange et al. [49, 78]	✓	-	46.2	50.5	34.8
DDT+ [49, 66]	✓	-	50.2	53.1	38.2
rOSD [49, 62]	✓	-	54.5	55.3	48.5
LOD [49, 63]	✓	-	53.6	55.1	48.5
DINO-seg [6, 49]		ViT-S/16 [14]	45.8	46.2	42.1
LOST [49]		ViT-S/16 [14]	61.9	64.0	50.7
DSS [37]		ViT-S/16 [14]	62.7	66.4	56.2
TokenCut		ViT-S/16 [14]	**68.8** (↑ **6.1**)	**72.1** (↑ **5.7**)	**58.8** (↑ **2.6**)
LOD + CAD* [49]	✓	-	56.3	61.6	52.7
rOSD + CAD* [49]	✓	-	58.3	62.3	53.0
LOST + CAD* [49]		ViT-S/16 [14]	65.7	70.4	57.5
TokenCut + CAD* [49]		ViT-S/16 [14]	**71.4** (↑ **5.7**)	**75.3** (↑ **4.9**)	**62.6** (↑ **5.1**)

* +CAD indicates to train a second stage class-agnostic detector with "pseudo-boxes" labels.

generate more precise boundaries for the mask. We have found that CRF usually provides the best results.

4 Experiments

We evaluated the suitability of TokenCut for three tasks: unsupervised single object discovery, unsupervised saliency detection and unsupervised video segmentation. We present implementation details in Section 4.1. The results of unsupervised single object discovery are shown in Section 4.2. The results for unsupervised saliency detection are presented in Section 4.3, and results for unsupervised video segmentation in Section 4.4. We provide ablation studies in Section 4.5.

4.1 Implementation details

Model details. For our experiments, we use the ViT-S/16 model [14] trained with self-distillation loss (DINO) [6] to extract features of patches. Following [49], we employ the key features of the last layer as the input features $\mathbf{v}$. Ablations on different features and ViT backbones are provided in Tab. 5. We set $\tau = 0.2$ for all image datasets and $\tau = 0.3$ for video datasets. The selection of τ is discussed in Section 4.5.

Algorithmic Cost. In terms of running time, our implementation takes approximately 0.32 seconds to detect a bounding box for a salient object region in a single image with resolution 480 × 480 using a single GPU QUADRO RTX 8000. Obtainaing a coarse mask from 20 frames of video with 320 x 576 resolution, requires an average of 30 seconds with standard deviation of around 4.5 seconds. Edge refinement the takes an additional 16.4 seconds on average with a standard deviation 1.4 seconds. As with single-frame graphs, the same video takes 0.93 seconds in average to obtain the coarse mask with standard deviation of 0.17 for all frames. The post processing step cost 16.1 seconds with standard deviation of 1.4. For n tokens, the algorithmic complexity for building such a graph is $O(n^2)$. Thus the average processing time grows with the square of the number of frames in the video.

Optical flow details. To generate optical flow, we use two different approaches: RAFT [54] and ARFlow [35]. The first one is supervised and the second one is self-supervised. We extract the optical flow at the original resolution of the image pairs, with the frame gaps n = 1 for DAVIS [40] and SegTV2 [31] dataset. For FBMS [39] we use n = 3 to compensate for the much slower rate of motion. This improves the optical flow quality as small pixel-level motions are hard to detect using off-the-shelf methods. Optical flow features are encoded as RGB values, using standard techniques for visualization of optical flow [3]. This allows us to directly use the pre-trained self-supervised transformers with optical flow encoded as RGB. Because of limits on available computational resources, we construct the video graph with a maximum of 90 frames on the DAVIS dataset. For videos longer than 90 frames, it is possible to aggregate results using non-overlapping subgraphs with maximum video frames of 90.

4.2 Unsupervised Single Object Discovery

Datasets. TokenCut has been evaluated on three commonly used benchmarks for unsupervised single object discovery: VOC07 [15], VOC12 [16] and COCO20K [34, 62]. VOC07 and VOC12 contain 5011 and 11540 images respectively which belong to 20 categories.

Table 2: **Comparisons for unsupervised saliency detection** We compare TokenCut to state-of-the-art unsupervised saliency detection methods on ECSSD [46], DUTS [65] and DUT-OMRON [73]. TokenCut achieves better results compared with other competitive approaches.

Method	ECSSD [46]			DUTS [65]			DUT-OMRON [73]		
	$maxF_\beta$(%)	IoU(%)	Acc.(%)	$maxF_\beta$(%)	IoU(%)	Acc.(%)	$maxF_\beta$(%)	IoU(%)	Acc.(%)
HS [71]	67.3	50.8	84.7	50.4	36.9	82.6	56.1	43.3	84.3
wCtr [79]	68.4	51.7	86.2	52.2	39.2	83.5	54.1	41.6	83.8
WSC [32]	68.3	49.8	85.2	52.8	38.4	86.2	52.3	38.7	86.5
DeepUSPS [38]	58.4	44.0	79.5	42.5	30.5	77.3	41.4	30.5	77.9
BigBiGAN [64]	78.2	67.2	89.9	60.8	49.8	87.8	54.9	45.3	85.6
E-BigBiGAN [64]	79.7	68.4	90.6	62.4	51.1	88.2	56.3	46.4	86.0
LOST [44, 49]	75.8	65.4	89.5	61.1	51.8	87.1	47.3	41.0	79.7
LOST [44, 49]+BS [5]	83.7	72.3	91.6	69.7	57.2	88.7	57.8	48.9	81.8
DSS [37]	-	73.3	-	-	51.4	-	-	56.7	-
TokenCut	80.3	71.2	91.8	67.2	57.6	90.3	60.0	53.3	88.0
TokenCut + BS [5]	**87.4** (↑ **3.7**)	77.2	93.4	75.5	62.4	91.4	**69.7** (↑ **11.9**)	61.8	89.7
TokenCut + CRF [29]	**87.4** (↑ **3.7**)	**77.7** (↑ **4.4**)	**93.6** (↑ **2.0**)	**75.7** (↑ **6.0**)	**62.8** (↑ **5.6**)	**91.5** (↑ **2.8**)	69.2	**61.9** (↑ **5.2**)	**89.8** (↑ **8.0**)

COCO20K consists of 19817 randomly chosen images from the COCO2014 dataset [34]. VOC07 and VOC12 are commonly used to evaluate unsupervised object discovery [11, 61–63, 66]. COCO20K is a popular benchmark for a large scale evaluation [62].

Evaluation metric. In line with previous research [11, 12, 50, 61–63, 66], we report performance using the *CorLoc* metric for precise localization. We use a single predicted bounding box for each image. For target images, CorLoc is 1.0 if the intersection over union (IoU) score between the predicted bounding box and the ground truth bounding boxes is superior to 0.5.

Quantitative Results. We evaluate the CorLoc scores in comparison with previous state-of-the-art single object discovery methods [27, 49, 57, 62, 63, 66, 78, 80] on VOC07, VOC12, and COCO20K datasets. These methods can be roughly divided into two groups according to whether the model uses information from the entire dataset or explores inter-image similarities. Because of the quadratic complexity of region comparison among images, models with inter-image similarities are generally difficult to scale to larger datasets. The selective search [57], edge boxes [80], LOST [49] and TokenCut do not require inter-image similarities and are thus much more efficient. As shown in the Tab. 1, TokenCut consistently outperforms all the previous methods on all the datasets by a large margin. Particularly, TokenCut ouperforms DSS [37] by 6.1%, 5.7% and 2.6% for VOC07, VOC12 and COCO20K respectively using the same ViT-S/16 features.

We also list a set of results that includes using a second stage unsupervised training strategy to boost the performance. This is referred to as Class-Agnostic Detection (CAD) and proposed in LOST [49]. For this, we first compute K-means on all the boxes produced by the first stage single object discovery model to obtain pseudo labels of the bounding boxes. Then a classical Faster RCNN [42] is trained on the pseudo labels. As shown in Tab. 1, TokenCut with CAD outperforms the state-of-the-art by 5.7%, 4.9% and 5.1% on VOC07, VOC12 and COCO20k respectively.

Qualitative Results. In Fig. 3, we provide visualization for LOST [49], DSS [37] and TokenCut*. For each method, we visualize the heatmap that is used to perform object detection. For LOST, the detection is mainly based on the map of inverse degree ($\frac{1}{\mathbf{d}_i}$). For DSS, the heatmap is the attention map associated to the second eigenvector. For TokenCut, we display the second smallest eigenvector. The visual results demonstrate that TokenCut can extract a high quality segmentation for the salient object. Compared with LOST and DSS, TokenCut is able to extract a more complete segmentation as can be seen in the first and the second samples in Fig. 3. In other cases, when LOST and DSS are unable to detect a large object, TokenCut can detect the object properly. Examples for this can be seen in the third and fourth samples in Fig. 3.

Internet Images. We further tested TokenCut on Internet images*. The results are in Fig 5. It can be seen that even though the input images have noisy backgrounds, TokenCut can provide a precise attention map to cover the object and lead to an accurate prediction of the bounding box, demonstrates robustness of the method.

4.3 Unsupervised Saliency detection

Datasets. We validated the performance of TokenCutfor unsupervised Saliency detection using three datasets : Extended Complex Scene Saliency Dataset(ECSSD) [46], DUTS [65] and DUT-OMRON [73]. ECSSD contains 1 000 real-world images of complex scenes for testing. DUTS contains 10 553 train and 5 019 test images. The training set is collected from the ImageNet detection train/val set. The test set is collected from ImageNet test, and the SUN dataset [70]. Following the previous work [44], we report the performance on the DUTS-test subset. DUT-OMRON [73] contains 5 168 images of high quality natural images for testing.

*More visual results can be found in the project webpage.

*We provide an online demo allowing to test Internet images.

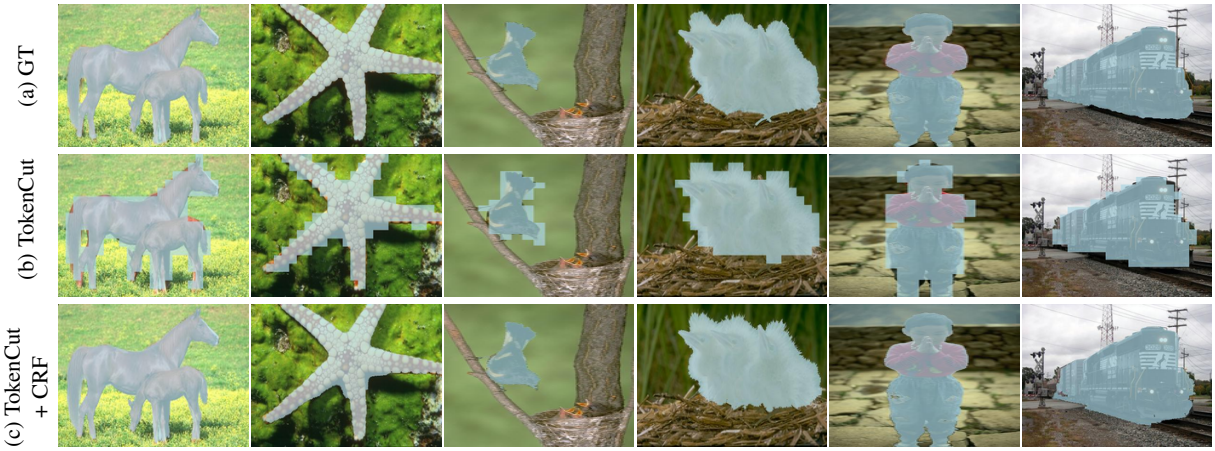
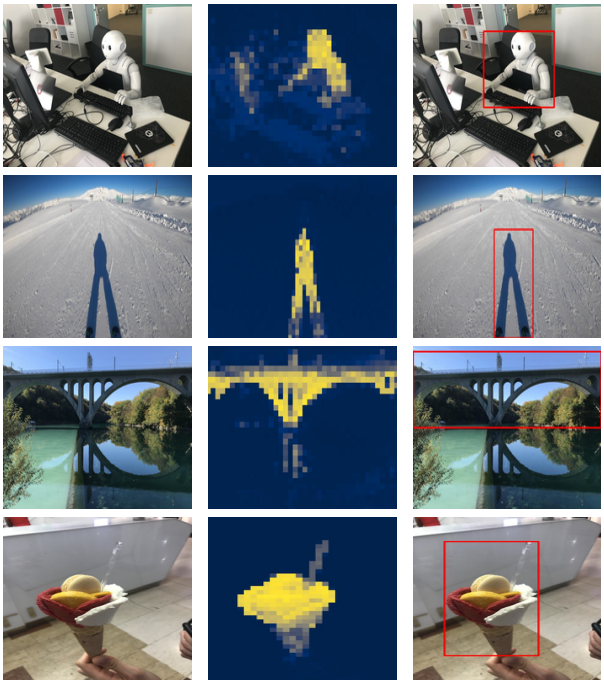


Figure 4: **Visual results of unsupervised segments on ECSSD [46].** In (a), we show the ground truth. (b) is TokenCut coarse mask segmentation result. The performance of TokenCut + Bilateral Solver (BS) is presented in (c).



(a) Input (b) Eigen Attention (c) Detection

Figure 5: **Visualization of images from the Internet.** We show the input images, our eigen attention, and final detection in (a), (b), and (c) respectively.

Evaluation Metrics. We report three standard metrics: F-measure, IoU and Accuracy. *F-measure* is a standard measure for saliency detection, computed as $F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall}$, where the Precision and Recall are defined using a binarized predicted mask and a ground truth mask. The $maxF_\beta$ is the maximum value of 255 uniformly distributed binarization thresholds. Following previous work [44, 64], we set $\beta = 0.3$ for consistency. *IoU*(Intersection over Union) score is computed based on the binary predicted mask and the ground-truth, the threshold is set to 0.5. *Accuracy* measures the proportion of pixels that have been correctly assigned to the object/background. The binarization threshold is set to 0.5 for masks.

Table 3: **Comparisons for unsupervised video segmentation**. We report *Jaccard index* and compare TokenCut to state-of-the-art unsupervised video segmentation methods on DAVIS [40], FBMS [39] and SegTV2 [31]. TokenCut achieves results that are similar to competing approaches.

Method	Flow	Training	DAVIS [40]	FBMS [39]	SegTV2 [31]
ARP [28]	CPM [22]		76.2	59.8	57.2
ELM [30]	Classic+NL [51]		61.8	61.6	-
MG [72]	RAFT [54]	✓	68.3	53.1	58.2
CIS [75]	PWCNet [52]	✓	71.5	63.5	62
DyStaB [74]⋆	PWCNet [52]	✓	**80.0**	**73.2**	**74.2**
DeSprite [76]‡	RAFT [54]	✓	79.1	71.8	72.1
TokenCut	RAFT [54]		64.3	60.2	59.6
TokenCut + BS [5]	RAFT [54]		75.1	61.2	56.4
TokenCut + CRF [29]	RAFT [54]		76.7	66.6	61.6
TokenCut	ARFlow [35]		62.0	61.0	58.9
TokenCut + BS [5]	ARFlow [35]		73.1	64.7	54.6
TokenCut + CRF [29]	ARFlow [35]		74.4	69.0	60.8

⋆: [74] is trained on DAVIS and evaluated on FBMS and SegTV2;
‡: [76] is optimized for each video separately.

Results. Qualitative results are shown in Tab. 2. TokenCut significantly outperforms previous state-of-the-art methods. Adding BS [5] or CRF [29] refines the boundary of an object and further boosts the TokenCut performance, as can be seen in the visual results presented in Fig. 4.

4.4 Unsupervised Video Segmentation

Datasets. We further evaluate TokenCut using three commonly used datasets for unsupervised video segmentation: DAVIS [40], FBMS [39] and SegTV2 [31]. DAVIS contains 50 high-resolution real-word videos, where 30 are for training and 20 are for validation. Pixel-

Figure 6: **Visual results of unsupervised video segmentation on DAVIS [40].** In (a), we show the ground truth segmentation. For TokenCut, we illustrate its coarse mask in (b) and refinement results with CRF in (c).

wise annotations are depicted for the principle moving object within the scene for each frame. FBMS consists of 59 multiple moving object videos, providing 30 videos for testing with a total of 720 annotation frames. SegTV2 contains 14 full pixel-level annotated video for multiple objects segmentation. Following [72], we fuse the annotation of all moving objects into a single mask on FBMS and SegTV2 datasets for fair comparison.

Evaluation metrics. We report performance using *Jaccard index*. The Jaccard index measures the intersection of union between an output segmentation M and the corresponding ground-truth mask G, which has been formulated as $\mathcal{J} = \frac{|M \cap G|}{|M \cup G|}$.

Results. We compare TokenCut to the state-of-the art unsupervised video segmentation results in Tab. 3. TokenCut achieves competitive performances for this task. Note that DyStaB [74] must be trained on the entire DAVIS training set and uses the pretrained model for evaluation with the FBMS and SegTV2 datasets. DeSprite [76] learns an auto-encoder model to optimize on each individual video. In contrast, TokenCut does not require training and generalizes well for all three datasets. Visual results are illustrated in Fig. 6, TokenCut can precisely segment moving objects even in the case of challenging occlusions. Adding CRF as a post-processing further improves the boundary for segmented regions*.

4.5 Analysis

Impact of τ. In Tab. 4, we provide an analysis on τ defined in Eqn 4. The results indicate that the effects of variations in τ value are not significant and that a suitable threshold is $\tau = 0.2$ for image input and $\tau = 0.3$ for video input.

*The segmentation results of entire videos can be found in the project webpage.

Table 4: **Analysis of τ.** We report CorLoc for unsupervised single object discovery on VOC07 [15], VOC12 [16] and COCO20K [34, 62] datasets, and Jacard index on DAVIS [40].

τ	CorLoc VOC07 [15]	CorLoc VOC12 [16]	CorLoc COCO20K [34, 62]	Jaccard Index DAVIS [40]
0	67.4	71.3	56.1	70.7
0.1	68.6	72.1	58.2	74.6
0.2	**68.8**	**72.1**	**58.8**	75.8
0.3	67.7	72.1	58.2	**76.7**

Table 5: **Analysis of different backbones.** We report CorLoc for unsupervised single object discovery on VOC07 [15], VOC12 [16] and COCO20K [34, 62] datasets.

Method	Backbone	VOC07 [15]	VOC12 [16]	COCO20K [34, 62]
LOST [49]	DINO-S/16 [6, 14]	61.9	64.0	50.7
TokenCut	DeiT-S/16 [14, 56]	2.39	2.9	3.5
TokenCut	MoCoV3-S/16 [7, 14]	66.2	66.9	54.5
TokenCut	DINO-S/16 [6, 14]	**68.8** (↑ **6.9**)	72.1 (↑ **8.1**)	58.8 (↑ **8.1**)
LOST [49]	DINO-S/8 [6, 14]	55.5	57.0	49.5
TokenCut	DINO-S/8 [6, 14]	67.3 (↑ **11.8**)	71.6 (↑ **14.6**)	**60.7** (↑ **11.2**)
LOST [49]	DINO-B/16 [6, 14]	60.1	63.3	50.0
TokenCut	MAE-B/16 [14, 19]	61.5	67.4	47.7
TokenCut	DINO-B/16 [6, 14]	68.8 (↑ **8.7**)	**72.4** (↑ **9.1**)	59.0 (↑ **9.0**)

Backbones. In Tab. 5, we provide an ablation study with different transformer backbones. The *"-S"* and *"-B"* are ViT small [6, 14] and ViT base [6, 14] architecture respectively. The *"-16"* and *"-8"* represents patch sizes 16 and 8 respectively. The *"DeiT"* is pre-trained supervised transformer model. The *"MoCoV3"* [7] and *"MAE"* [19] are pre-trained self-supervised transformer model. We optimise τ for different backbones: τ is set to 0.3 for MoCov3 and MAE, while for DINO and Deit τ is set to 0.2. Several insights can be found: *1)* TokenCut is not suitable for supervised transformer models, while self-supervised transformers provide more powerful

Table 6: **Analysis of different bi-partition methods.** We report CorLoc for unsupervised single object discovery on VOC07 [15], VOC12 [16] and COCO20K [34, 62] datasets.

Bi-partition	VOC07	VOC12	COCO20K
Mean	**68.8**	**72.1**	58.8
Energy (Eqn 1)	67.3	69.7	-
EM	63.0	65.7	59.3
K-means	67.5	69.2	**61.6**

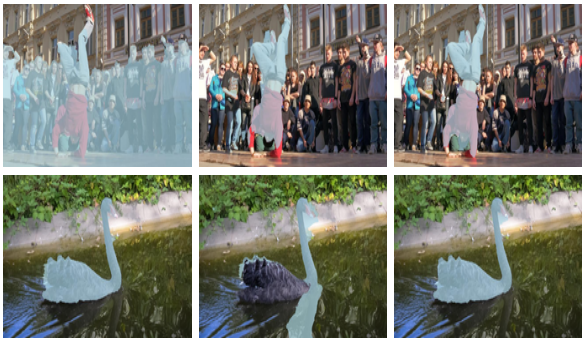
Table 7: **Analysis of video input.** We report Jaccard index for video segmentation on DAVIS [40], FBMS [39] and SegTV2 [31] with using input. "RGB + Flow" refers to using both video RGB frame and RGB representation of optical flow as input to the vision transformer. "Mean_Flow" indicates using mean of flow instead of flow features extracted from vision transformer". "RGB" and "Flow" present using only either RGB frames or optical flow as input. "CRF" indicates whether edge refinement step using CRF [29] is computed.

Input	CRF	DAVIS [40]	FBMS [39]	SegTV2 [31]
Graph per frame				
RGB	✓	62.4	67.2	61.0
Flow	✓	64.1	52.8	53.7
RGB + Flow	✓	76.4	63.2	**64.4**
Graph per video				
RGB		51.8	58.4	59.3
Flow		49.9	48.3	46.7
RGB + Flow		64.3	60.2	59.6
RGB	✓	62.2	**67.5**	63.7
Flow	✓	63.1	50.2	50.2
RGB + Mean_Flow	✓	37.5	23.3	15.7
RGB + Flow	✓	**76.7**	66.6	61.6

features allowing completing the task with TokenCut. *2)* As LOST [49] relies on a heuristic seeds expansion strategy, the performance varies significantly using different backbones. While our approach is more robust. Moreover, as no training is required for TokenCut, it might be a more straightforward evaluation for the self-supervised transformers.

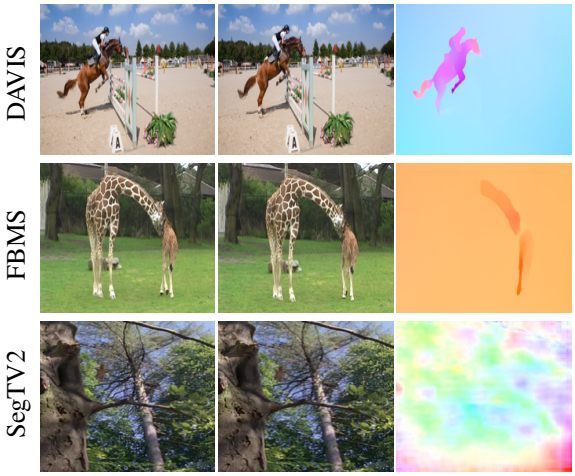
Bi-partition strategies. In Tab. 6, we study different strategies to separate the nodes in into two groups using the second smallest eigenvector. We consider three natural methods: mean value (Mean), Expectation-Maximisation (EM), K-means clustering (K-means). We have also tried to search for the splitting point based on the best Ncut energy (Eqn 1). Note this approach is computationally expensive due to the quadratic complexity. The result suggests that the simple mean value as the splitting point performs well for most cases.

Video input. We also study the impact of using RGB or optical Flow for video segmentation. Quantitative results are presented in Tab. 7. We can see constructing graph on the entire video is better than constructing the



(a) RGB (b) Flow (c) RGB + Flow

Figure 7: **Visualization on DAVIS [40] using different inputs.** We show segmentation results with RGB, Flow and RGB + Flow in (a), (b) and (c) respectively.



(a) Frame t (b) Next frame (c) Flow

Figure 8: **Visual results of optical flow in DAVIS, FBMS, SegTV2**. In (a), we show Frame t. (b) is the following frame (t + 3 for FBMS and t + 1 for SegTV2) used to compute the optical flow. (c) is the optical flow using RAFT [54]. Note that optical flow in the DAVIS dataset has high quality, whereas the flow in the FBMS and SegTV2, can sometimes fail, as in the cases shown in rows 2 and 3.

graph per frame. We show an anaylsis by using the mean of optical flow feature and use it directly without feeding into transformer. The results illustrate that constructing the graph with RGB and RGB representation of the Flow together can significantly improve the performances over DAVIS [40]. On FBMS [39] and SegTV2 [31], due to the low quality of optical flow, the motion of salient objects are not detected in the optical flow as can be seen from the fact that they are not visible in the RGB visualisation. Some failure cases are shown in Fig. 8. This failure of optical flow to detect slow motion impedes the inference process for augmenting appearance with optical flow features. The low quality of optical flow can be attributed to three factors: 1) small motion between two

Table 8: **Analysis of video graph.** We report Jaccard index ($\mathcal{J}$) for video segmentation on DAVIS [40] with different video graphs. "single frame" represent creating the graph for each frame separately.

Nodes	Edges	DAVIS ($\mathcal{J}$)
Video	$\min(S(\mathbf{v}_i^I, \mathbf{v}_j^I), S(\mathbf{v}_i^F, \mathbf{v}_j^F))$	73.7
Video	$\max(S(\mathbf{v}_i^I, \mathbf{v}_j^I), S(\mathbf{v}_i^F, \mathbf{v}_j^F))$	71.1
Video	$\frac{S(\mathbf{v}_i^I, \mathbf{v}_j^I)+S(\mathbf{v}_i^F, \mathbf{v}_j^F)}{2}$	**76.7**
Single Frame	$\frac{S(\mathbf{v}_i^I, \mathbf{v}_j^I)+S(\mathbf{v}_i^F, \mathbf{v}_j^F)}{2}$	76.4

frames; 2) low quality of raw image, for instance several examples in SegTV2, such as birdfall; 3) the absence of fine-tuning for the pre-trained optical flow model on these three datasets. Using both RGB appearance and flow lead to a slight improvement before edge refinement, but slightly worse results after edge refinement compared to using only RGB appearance. Some qualitative results are illustrated in Fig. 7. We can see how RGB frame and optical flow are complementary to each other: in the first row, the target moving person shares semantically similar features to other audiences and using only RGB frames would produce a mask cover all the persons; in the second row, the flow also has non-negligible values on the surface of the river, thus using only flow leads to worse performance.

Video graph. In Tab. 8, we provide an analysis for different ways to construct graphs for video. For edges, we also consider the minimum and maximum values between the flow and RGB similarities. For nodes, a natural baseline is to build a graph for each single frame. We can see that the optimal choice is to use the average value of the flow and RGB similarities (Eqn. 4) and build a graph for an entire video.

5 Discussion

Multi-Object Segmentation. In the context of the Unsupervised Single Object Discovery task, the primary objective is to identify the most salient object within a given image. Consequently, we only choose the largest connected component in our approach. However, TokenCutcan identify more than one connected component in the second smallest eigenvector when multiple objects are present in the images. To illustrate this capability, we have included two examples in Fig 10. In Fig 11, we provide examples when multiple objects are moving from different directions. These results illustrate the robustness of our method.

Limitations. Despite the good performance of the TokenCut proposal, it has several limitations. We show several failure cases in Fig. 9: i) As seen in the 1st row, TokenCut focuses on the largest salient part in the image, which may not be the desired object. ii) Similar to LOST [49], TokenCut assumes that a single salient object occupies the foreground. If multiple overlapping objects are present in an image, both LOST and our approach would fail to detect one of the object, as displayed in the 2nd row. iii) For object detection, neither LOST norTokenCut can handle occlusion properly, as shown in the 3rd row.

(a) LOST Inverse Attn. (b) LOST Detection (c) Our Eigen Attention (d) Our Detection

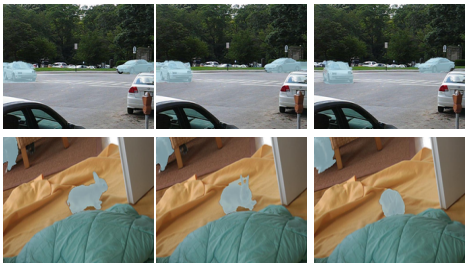
Figure 9: **Failure cases on VOC12 (1st and 2nd row) and COCO (3rd row).** LOST [49] mainly relies on the map of inverse degrees (a) to perform detection (b). For our approach, we illustrate the eigenvector in (c) and our detection in (d). Blue and Red bounding boxes indicate the ground-truth and the predicted bounding boxes respectively.

(a) Raw image (b) Eigen Attention (c) Coarse mask (d) Fine mask

Figure 10: **Visual results of multiple objects in images**. In (a), we show the original image. TokenCut eigen Attention is illustrated in (b). (c) is TokenCut coarse mask segmentation results before selecting the largest connected component. (d) is TokenCut fine mask using bilateral solver.

6 Conclusion

This paper describes TokenCut, an unified and effective approach for both image and video object segmentation without the need for supervised learning. TokenCut uses



(a) Frame t (b) Frame t+5 (c) Frame t+10

Figure 11: **Visual results of multiple objects moving.** We visualize the case when multiple object moving from different direction, TokenCut is also capable of segmenting the moving objects.

features from self-supervised transformers to constructs a graph where nodes are patches and edges represent similarities between patches. For videos, optical flow is incorporated to determine moving objects. We show that salient objects can be directly detected and delimited using the Normalized Cut algorithm. We evaluated this approach on unsupervised single object discovery, unsupervised saliency detection, and unsupervised video object segmentation, demonstrating that TokenCut can provide a significant improvement over previous approaches. Our results demonstrate that self-supervised transformers can provide a rich and general set of features that may likely be used for a variety of computer vision problems.

Acknowledgment

This work has been partially supported by the MIAI Multidisciplinary AI Institute at the Univ. Grenoble Alpes (MIAI@Grenoble Alpes - ANR-19-P3IA-0003), and by the EU H2020 ICT48 project Humane AI Net under contract EU #952026.

References

[1] Hamed H Aghdam, Abel Gonzalez-Garcia, Joost van de Weijer, and Antonio M López. Active learning for deep detection neural networks. In *ICCV*, 2019. 1

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv*, 2016. 4

[3] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011. 4, 6

[4] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv*, 2021. 2

[5] Jonathan T Barron and Ben Poole. The fast bilateral solver. In *ECCV*, 2016. 2, 3, 5, 7, 8

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 6, 9

[7] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 2, 9

[8] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *CVPR*, 2021. 1

[9] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Show, match and segment: Joint weakly supervised learning of semantic matching and object co-segmentation. *PAMI*, 2020. 2

[10] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *TPAMI*, 2014. 3

[11] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*, 2015. 2, 7

[12] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. In *ECCV*, 2010. 7

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2018. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 2, 3, 6, 9

[15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. PASCAL VOC2007. 2, 6, 9, 10

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. PASCAL VOC2012. 2, 6, 9, 10

[17] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In *CVPR*, 2022. 2

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 2013. 1

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *CVPR*, 2022. 2, 9

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv*, 2015. 2

[21] Kuang-Jui Hsu, Yen-Yu Lin, Yung-Yu Chuang, et al. Co-attention cnns for unsupervised object co-segmentation. In *IJCAI*, 2018. 2

[22] Yinlin Hu, Rui Song, and Yunsong Li. Efficient coarse-to-fine patchmatch for large displacement optical flow. In *CVPR*, 2016. 8

[23] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, 2013. 3

[24] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010. 2

[25] Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *CVPR*, 2012. 2

[26] Tsung-Wei Ke, Jyh-Jing Hwang, and Stella X Yu. Universal weakly supervised segmentation by pixel-to-segment contrastive learning. *ICLR*, 2021. 1

[27] Gunhee Kim and Antonio Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *NeurIPS*, 2009. 6, 7

[28] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, 2017. 3, 8

[29] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 2011. 2, 3, 5, 7, 8, 10

[30] Dong Lao and Ganesh Sundaramoorthi. Extending layered models to 3d motion. In *ECCV*, 2018. 3, 8

[31] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 2, 6, 8, 10

[32] Nianyi Li, Bilin Sun, and Jingyi Yu. A weighted sparse coding framework for saliency detection. In *CVPR*, 2015. 3, 7

[33] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. In *NeurIPS*, 2021. 2

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 2, 6, 7, 9, 10

[35] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *CVPR*, 2020. 6, 8

[36] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *ICLR*, 2021. 1

[37] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *CVPR*, 2022. 2, 5, 6, 7

[38] Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox. Deepusps: Deep robust unsupervised saliency prediction via self-supervision. In *NeurIPS*, 2019. 3, 7

[39] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *TPAMI*, 2013. 2, 6, 8, 10

[40] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 2, 6, 8, 9, 10, 11

[41] Alex Pothen, Horst D Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM journal on matrix analysis and applications*, 1990. 4

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NIPS*, 2015. 7

[43] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Yong Jae Lee, Alexander G Schwing, and Jan Kautz. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In *CVPR*, 2020. 1

[44] Xi Shen, Alexei A Efros, Armand Joulin, and Mathieu Aubry. Learning co-segmentation by segment swapping for retrieval and discovery. *arXiv preprint arXiv:2110.15904*, 2021. 7, 8

[45] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *TPAMI*, 2000. 2, 3, 4

[46] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *TPAMI*, 2015. 2, 7, 8

[47] Gyungin Shin, Samuel Albanie, and Weidi Xie. Unsupervised salient object detection with spectral cluster voting. In *CVPRW*, 2022. 3

[48] Yawar Siddiqui, Julien Valentin, and Matthias Nießner. Viewal: Active learning with viewpoint entropy for semantic segmentation. In *CVPR*, 2020. 1

[49] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *BMVC*, 2021. 2, 3, 5, 6, 7, 9, 10, 11

[50] Parthipan Siva, Chris Russell, Tao Xiang, and Lourdes Agapito. Looking beyond the image: Unsupervised learning for object saliency and detection. In *CVPR*, 2013. 7

[51] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 8

[52] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 8

[53] Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014. 2

[54] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 6, 8, 10

[55] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv*, 2022. 2

[56] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 9

[57] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 2013. 6, 7

[58] Charles F Van Loan and G Golub. Matrix computations. *The Johns Hopkins University Press*, 1996. 4

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3, 4

[60] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, 2011. 2

[61] Huy V Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez, and Jean Ponce. Unsupervised image matching and object discovery as optimization. In *CVPR*, 2019. 2, 7

[62] Huy V Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *ECCV*, 2020. 2, 6, 7, 9, 10

[63] Huy V Vo, Elena Sizikova, Cordelia Schmid, Patrick Pérez, and Jean Ponce. Large-scale unsupervised object discovery. *arXiv*, 2021. 2, 6, 7

[64] Andrey Voynov, Stanislav Morozov, and Artem Babenko. Object segmentation without labels with large-scale generative models. In *ICML*, 2021. 3, 7, 8

[65] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017. 2, 7

[66] Xiu-Shen Wei, Chen-Lin Zhang, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Unsupervised object discovery and co-localization by deep descriptor transformation. *Pattern Recognition*, 2019. 6, 7

[67] Yichen Wei, Fang Wen, Wangjiang Zhu, and Jian Sun. Geodesic saliency using background priors. In *ECCV*, 2012. 3

[68] Bichen Wu, Forrest Iandola, Peter H Jin, and Kurt Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *CVPRW*, 2017. 1

[69] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *CVPR*, 2018. 1

[70] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 7

[71] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, 2013. 3, 7

[72] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *CVPR*, 2021. 3, 8, 9

[73] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013. 2, 3, 7

[74] Yanchao Yang, Brian Lai, and Stefano Soatto. Dystab: Unsupervised object segmentation via dynamic-static bootstrapping. In *CVPR*, 2021. 3, 8, 9

[75] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 3, 8

[76] Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. Deformable sprites for unsupervised video decomposition. *CVPR*, 2022. 3, 8, 9

[77] Jing Zhang, Tong Zhang, Yuchao Dai, Mehrtash Harandi, and Richard Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *CVPR*, 2018. 3

[78] Runsheng Zhang, Yaping Huang, Mengyang Pu, Jian Zhang, Qingji Guan, Qi Zou, and Haibin Ling. Object discovery from a single unlabeled image by mining frequent itemsets with multi-scale features. *TIP*, 2020. 6, 7

[79] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *CVPR*, 2014. 3, 7

[80] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 6, 7