Self-Supervised Video Copy Localization with Regional Token Representation

Minlong Lu, Yichen Lu, Siwei Nie, Xudong Yang, and Xiaobo Zhang

Ant Group, Hangzhou, China
{luminlong.lml,luyichen.lyc,jiegang.yxd,ayou.zxb}@antgroup.com

Abstract. The task of video copy localization aims at finding the start and end timestamps of all copied segments within a pair of untrimmed videos. Recent approaches usually extract frame-level features and generate a frame-to-frame similarity map for the video pair. Learned detectors are used to identify distinctive patterns in the similarity map to localize the copied segments. There are two major limitations associated with these methods. First, they often rely on a single feature for each frame, which is inadequate in capturing local information for typical scenarios in video copy editing, such as picture-in-picture cases. Second, the training of the detectors requires a significant amount of human annotated data, which is highly expensive and time-consuming to acquire. In this paper, we propose a self-supervised video copy localization framework to tackle these issues. We incorporate a Regional Token into the Vision Transformer, which learns to focus on local regions within each frame using an asymmetric training procedure. A novel strategy that leverages the Transitivity Property is proposed to generate copied video pairs automatically, which facilitates the training of the detector. Extensive experiments and visualizations demonstrate the effectiveness of the proposed approach, which is able to outperform the state-of-the-art without using any human annotated data.

Keywords: Self-Supervised Video Copy Localization · Regional Feature

1 Introduction

With the exponential growth of content uploaded to video sharing platforms such as TikTok and YouTube, the issue of unauthorized use and distribution of copyrighted material has become more serious. It is non-trivial to identify copied videos, as there are numerous content modifications that can be applied, and users may intentionally make adversarial edits to evade moderation systems [19, 30]. As a result, video copy localization is becoming increasingly important. Beyond detecting the presence of copied content at the video level [31], video copy localization precisely identifies the start and end timestamps of all the copied segments, which is intuitive and valuable in applications such as copyright protection, video filtering and recommendation [15].

Existing video copy localization methods usually start by extracting frames at specific time intervals from the input videos. Frame-level features are then

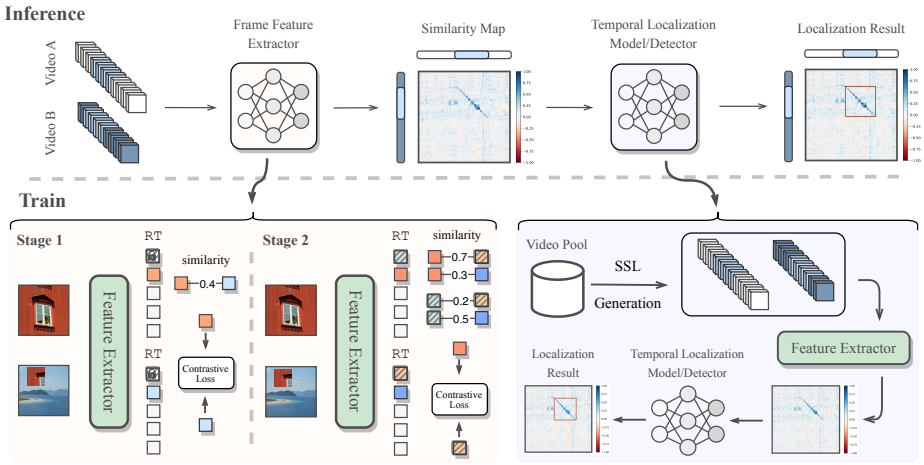

Fig. 1: Our Self-Supervised Video Copy Localization Framework. We incorporate a Regional Token (RT) to the ViT model to build our feature extractor, which is trained in a two-stage asymmetric process. Our temporal localization model/detector is trained using generated copied video pairs in a self-supervised manner.

extracted to generate a frame-to-frame similarity map for the video pair. Finally, temporal localization models are used to identify distinctive patterns of the copied segments in the similarity map, which localize them and output the respective time ranges [18,19,22]. While these methods have achieved good performance, there are two major limitations. First, they often rely on a global feature for each frame, which is inadequate for typical scenarios in video copy editing. For example, in cases of camcording or in picture-in-picture scenarios, the copyrighted content might only occupy a small region within the entire frame. Using a single feature may not effectively capture the specific region of interest without being affected by the unrelated content in the rest of the frame [31]. Second, to deal with the wide variety of appearance of copied segments in the similarity map, these methods often employ object detectors that require a significant amount of human annotated data to train. However, acquiring such annotated data can be very expensive and time-consuming [19].

Two strategies are proposed to cope with picture-in-picture scenarios in the winning solutions of recent image and video copy detection competitions [11,30]. The first strategy involves generating multiple crops for the input and extracting features for each crop, which results in a feature set for each image or frame for similarity measurement. The crops can be human-defined regions [39] or generated via region proposals and detection-based techniques [28,39]. The second strategy concatenates two images as input to a Vision Transformer, which outputs a binary prediction indicating whether copied content exists [21]. This method implicitly handles copies in local regions through the patch tokens and the self-attention mechanism. While these strategies produce improved results, they require a significant increase in computational cost.

To address these issues, we propose a self-supervised framework for video copy localization, as illustrated in Fig. 1. To better capture local information in feature extraction, we make simple yet effective modifications to the architecture and training process of the Vision Transformer (ViT) [7]. A Regional Token is introduced into the ViT model, which provides an additional feature vector for each frame. The feature extractor is trained in a two-stage asymmetric procedure with a contrastive loss. Our model effectively extracts local features and produces improved performance, with only a minimal increase in parameters and computational cost. It is also noteworthy that our Regional Token learns to focus on picture-in-picture regions without the need for explicit supervision.

To alleviate the need for human-annotated data, we propose a novel strategy to generated video pairs automatically, which are used to train our detector for temporal localization in a self-supervised manner. In the generation process, video segments are randomly selected from source videos. These segments undergo spatial and temporal transformations and are then inserted into another video to create the "copy video". In addition, we utilize the Transitivity Property [19, 23] to expand the set of generated video pairs, which introduces a broader range of similarity patterns to the copied video pairs, leading to better performance in models trained on these data.

Our contributions can be summarized as follows:

- We propose a self-supervised approach for video copy localization, where our feature extractor and temporal localization model are both trained in a self-supervised manner. Without using any human-annotated data, our model achieves state-of-the-art performance on the VCSL and VCDB datasets.
- We make an interesting discovery: by simply adding a new token and modifying the training process, our model can effectively learn regional representation and improve performance. The visualization of the attention weights for our Regional Token demonstrates a distinct focus on local regions.
- The performance of our model can be further improved by fine-tuning the detector using human-annotated data. With only 1% of the VCSL training data, our model outperforms the supervised training model that uses the entire training set. The fully fine-tuned model achieves a segment-level F1 score of 71.56% and a video-level F1 score of 96.46% on the VCSL dataset, surpassing previous best-performing methods by a significant margin.

2 Related Work

2.1 Image and Video Copy Detection

The task of image copy detection is to determine whether two images are derived from the same original source, potentially altered by image editing. Both traditional descriptors [8, 24, 43] and deep features [42] are explored for image copy detection. Recently, self-supervised learning approaches achieve impressive performance [31,41]. These methods are well-suited for this task, as the data augmentation used to generate positive pairs resembles the process of "copy editing", making the training objective consistent with copy detection [11].

Video copy detection methods aim to determine whether any copied segments exist in a pair of videos [2, 20, 25–27]. In addition to a video-level label, video copy localization provides the start and end timestamps for all copied segments in a pair of untrimmed videos [10, 15, 19]. Existing methods follow a common pipeline [19], where temporal localization models are utilized to identify distinctive patterns in the similarity map to localize copied segments [18]. The similarity maps are usually constructed using frame-level features [25,35,41]. For temporal localization, traditional methods such as Hough Voting [9], Dynamic Programming [5], Dynamic Time Warping [3], and Temporal Network [34] have been employed. Object detection models are also used to localize copied segments in the similarity map, which learn to account for the diverse appearance of copied segments from a large amount of human-annotated video pairs [19], and are able to produce more precise localization [18, 22].

2.2 Self-supervised Learning for Image and Video

Early self-supervised learning approaches for image and video focus on the design of pretext tasks, such as relative position prediction [6] and frame order prediction [29,40]. Contrastive methods then become prevalent in self-supervised learning [4,14,17,32,33], which learn by pulling positive pairs closer and pushing negative samples apart. Masked image modeling also achieves impressive performance, which learns by masking some patches of the input images and then reconstructing them from the visible ones [1, 12, 16, 36].

Although self-supervised learning is popular in the video domain, its applications to video copy detection and localization are still uncommon [20, 26]. The concept of self-supervision is used in [2, 15] to generate copied video pairs for video copy localization. However, the data augmentation strategies are relatively straightforward, leading to preliminary results. Currently, state-of-the-art performance is still achieved by models trained on human-annotated datasets [18]. In our work, we incorporate Transitivity Propagation [23] for self-supervised data augmentation, which introduces more diverse similarity patterns to the copied video pairs, leading to better performance in models trained on these data.

3 Method

Our self-supervised video copy localization framework contains two key components: a frame feature extractor, which incorporates a Regional Token into the ViT architecture to capture local information, and a temporal localization model, which is an object detector trained with a self-supervised data augmentation. In this section, we first give the problem definition of video copy localization, and then introduce our feature extractor and temporal localization model.

3.1 Problem Definition of Video Copy Localization

We denote a query video as $v^q = \{f_i^q\}_{i=1}^Q$ and a reference video as $v^r = \{f_j^r\}_{j=1}^R$, where Q and R are the numbers of extracted frames, and f_i^q and f_j^r are the

i-th and j-th frames for the query and reference videos, respectively. If a frame subsequence $\{f_i^q\}_{i=s^q}^{e^q}$ in v^q and a frame subsequence $\{f_j^r\}_{j=s^r}^{e^r}$ in v^r are derived from the same original source, subjected to possible editing, they are considered as a pair of copied segments. This copied segment pair can be represented as $\{s^q, e^q, s^r, e^r\}$, where s^q and e^q are the start and end frame indices (or timestamps) of the segment in the query video v^q, and s^r and e^r are the corresponding indices in the reference video v^r. There might be one or more copied segment pairs in a copied video pair, which can be represented as a list of 4-tuples. The objective of video copy localization is to provide the exact start and end timestamps for all copied segments between the query video and the reference video.

3.2 Self-Supervsed Feature Extraction

To tackle typical video copy scenarios where the copyrighted content may occupy only a small portion of the entire frame, we make simple modifications to the architecture and training process of the Vision Transformer (ViT) [7] to serve as our feature extractor. We introduce an additional token, referred to as the Regional Token, into the ViT architecture. This token, $\mathbf{x}_{\text{region}}$, is concatenated with the CLS token and the patch tokens. Learnable position embeddings are then added to these token embeddings following [7]. The resulting sequence, denoted as $\mathbf{z}_0$, serves as the input to the multi-head self-attention (MSA) [7,38] and MLP layers. The final feature vectors of the Regional Token and CLS token are produced by a linear layer:

$$\mathbf{z}_0 = [\mathbf{x}_{\text{region}}; \mathbf{x}_{cls}; \mathbf{x}_1; \mathbf{x}_2; ...; \mathbf{x}_P] + \mathbf{E}_{pos}, \qquad \mathbf{E}_{pos} \in \mathbb{R}^{(P+2)\times D} \tag{1}$$

$$\mathbf{z}'_l = \text{MSA}(\text{LN}(\mathbf{z}_{l-1})) + \mathbf{z}_{l-1}, \qquad l = 1, ..., L \tag{2}$$

$$\mathbf{z}_l = \text{MLP}(\text{LN}(\mathbf{z}'_l)) + \mathbf{z}'_l, \qquad l = 1, ..., L \tag{3}$$

$$\mathbf{f}_{\text{region}}, \mathbf{f}_{cls} = \text{Linear}(\mathbf{z}_L[0], \mathbf{z}_L[1]), \tag{4}$$

where L is the number of layers and $\mathbf{E}_{pos}$ is the position embedding.

The feature extractor is trained in a contrastive manner, where common copy editing augmentations are used to generate the positive samples [21, 31]. We adapt the NT-Xent loss [4] as the objective function, with the loss for a positive image pair (I_i, I_j) defined as follows:

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(I_i, I_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} \exp(\text{sim}(I_i, I_k)/\tau)}, \tag{5}$$

where $\mathbb{1}_{[k\neq i]}$ is an indicator function that equals 1 iff $k \neq i$, τ is the temperature parameter, and $\text{sim}(\cdot, \cdot)$ denotes the similarity between two samples. The final loss is computed across all positive pairs in the mini-batch of size N.

We employ a two-stage asymmetric training process for the feature extractor, as illustrated in Fig 1. In the first stage, we fix the Regional Token and our model learns general features using the CLS token. The similarity function $\text{sim}(\cdot, \cdot)$ is defined as the cosine similarity between the CLS token features $\mathbf{f}_{cls}$. In the second stage, we unlock the Regional Token for further training. The similarity

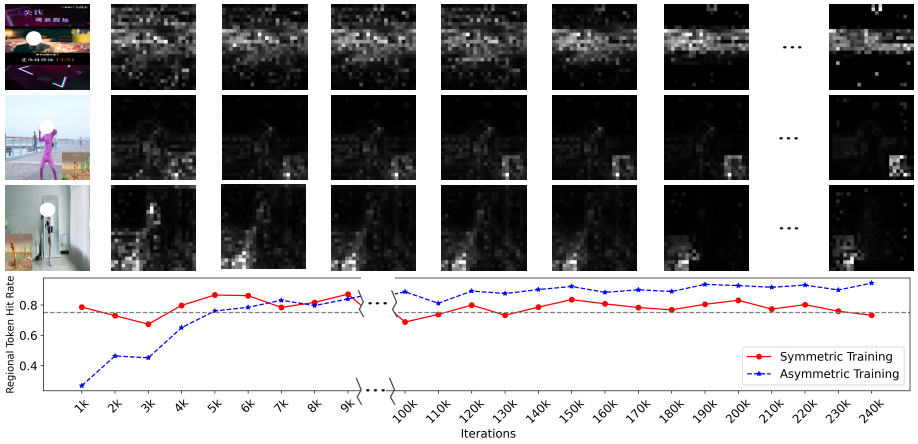

Fig. 2: Visualization of the training process. The upper part shows the evolution of the self-attention maps for the Regional Token. The bottom part compares the "Regional Token Hit Rate" for a symmetric training and our asymmetric training procedure.

function is adjusted to compute the maximum cosine similarity across all pairwise combinations of the $\mathbf{f}_{\text{region}}$ and $\mathbf{f}_{cls}$ features between the two images. This training strategy creates an asymmetry between the Regional Token and CLS token, which enables the Regional Token to focus on local regions and the CLS token to focus on global information, as demonstrated in Section 4.5.

Analysis. We conduct preliminary experiments to demonstrate the effectiveness of our asymmetric training. Figure 2 visualizes the training process of our model, with the upper part showing the evolution of the self-attention maps for the Regional Token. As training progresses, the attention gradually concentrates on the local regions of the picture-in-picture areas.

We also define a metric called the "Regional Token Hit Rate" to analyze the training process, which is the percentage of time when the similarity between two frames is obtained using at least one feature from the Regional Token. This metric is evaluated using 200 randomly selected VCSL testing video pairs that include picture-in-picture editing. In the second stage of our asymmetric training, the "Regional Token Hit Rate" begins at a low value, as shown in the bottom part of Figure 2. This indicates that the frame similarity is initially dominated by the CLS token feature, which learns to produce a global feature in the first stage. When the model encounters picture-in-picture data during training, leveraging local features will produce better similarity measures for the inputs, which in turn leads to a decrease in the objective function. This encourages our Regional Token to gradually focus on local regions as training progresses. As a result, the metric gradually increases to 94.36%, which reflects the learning process and indicates the increasing effectiveness of the Regional Token.

As a comparison, we also test a symmetric training process where the CLS token and the Regional Token are trained together in a single stage. The "Regional Token Hit Rate" for this method approximates 75%, aligning with the

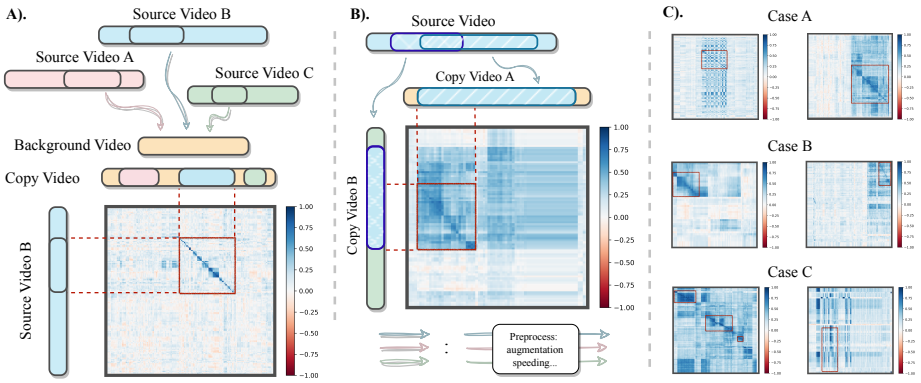

Fig. 3: Self-Supervised Copied Video Pair Generation. A) Vanilla Strategy: In this example, three source video segments are selected and inserted into the background video to create three copied video pairs. Only one pair is visualized in this figure for clarity. **B)** Transitivity Propagation: Two copy videos containing segments derived from the same source video with temporal overlap form a copied video pair. **C)** More examples of visualized similarity maps.

probability of random selection. Note that each frame has a CLS token feature and a Regional Token feature, and the similarity between two frames is computed as the maximum cosine similarity among the four possible combinations. Therefore, the expected "Regional Token Hit Rate" for random selection would be 3 out of 4, or 75%. This demonstrates that it is our asymmetric training process that enables the Regional Token to effectively learn the local information.

3.3 Self-Supervised Temporal Localization

Object detectors are employed for temporal localization in recent models [18,22], which learn from the diverse appearances of copied segments in the similarity map of human-annotated video pairs [19], producing more precise localizations than traditional methods. However, manual annotation is very expensive and time-consuming to acquire, posing a significant obstacle to scaling the training data. To tackle this issue, we generate copied video pairs automatically and train our temporal localization model/detector in a self-supervised manner. In the generation process, we use a vanilla strategy to generate copy videos and then use the Transitivity Property to extend the set of copied video pairs.

Vanilla Strategy. Given an arbitrary set of videos $\mathcal{V} = \{v_i\}_{i=1}^{N}$, we randomly select one background video v^b and several source videos $\mathcal{V}^s = \{v_1^s, v_2^s, \ldots, v_k^s\}$ to create a copy video. Video segments are randomly clipped from the source videos, undergo spatial and temporal augmentations, and are then inserted at a random timestamp in the background video, as illustrated in Fig. 3 (A). Temporal augmentation is performed by varying the frame sampling rate within a ratio of 0.5 to 2, resulting in either a slowdown or speedup of the original segment by a factor of up to 2. Spatial augmentation includes standard image transformations

(e.g. resized-crop, blur, color jitter, etc.), as well as video copy editing-styled transformations (e.g. superimpose objects or emojis on frames, overlay frames on a blurry background or other videos, etc.). Finally, the copy video v^c and the source videos in $\mathcal{V}^s$ form k copied video pairs, with the ground truth start and end timestamps being recorded during the augmentation process.

Transitivity Propagation. In real-life video copy scenarios, there are often semantic relations between the source and background videos, such as sharing the same topic or containing the same characters. However, this is not considered in the vanilla strategy, and the source video segments are often inserted into irrelevant background videos. In this case, the frames of the source segments exhibit higher similarity with each other than with the frames of the background videos. This leads to the appearance of a highlighted strip in the similarity map, which marks the start and end timestamps of the segments in one of the videos, as shown in Case A of Fig. 3 (C). The detector may take this additional hint as a shortcut to localize the copied segments, which might not generalize well to realistic scenarios. To mitigate this issue, we utilize the Transitivity Property [23] to expand the set of copied video pairs and enrich the data for the training of our detector.

The basic idea of the Transitivity Property is that if two copy videos contain segments derived from the same source video and these segments have temporal overlap, then the two videos form a copied video pair. The ground truth is the intersection of the segments, while the frames of the segments that are outside the intersection serve as a high similarity neighborhood. This produces an improved similarity map, where the pattern of the copied segments (e.g., a diagonal line) is embedded within a block of relatively high amplitude compared to the background, as shown in Fig. 3 (B) and Case B of Fig. 3 (C). Moreover, the segments in the two videos are augmented independently, which further increases the diversity of similarity patterns among the copied video pairs, leading to better performance in models trained on these data. In practice, we identify potential video pairs with intersecting segments by examining the copy video pool generated using the vanilla strategy. This process ensures that there is minimal cost associated with employing the Transitivity Property for generation.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our model on the VCDB [23] and VCSL [19] datasets. The VCSL dataset is currently the largest realistic dataset for video copy localization, consisting of over 160K user-generated copied video pairs with manually-labeled segment-level annotations. The VCSL testing set comprises 55,530 video pairs, including 27,765 copied pairs and an equal number of negative pairs serving as distractors. The VCDB dataset has a smaller scale, comprising 6,140 realistic copied video pairs with segment-level annotations. Our feature extractor is trained on the DISC dataset [11], which comprises 1M images designated for

copy detection training. We generate 602K video pairs from the VCSL training videos, including 240K positive pairs, to train our temporal localization model.
Evaluation Metrics. We follow the evaluation protocol in [19] and take the untrimmed video pairs as input to identify all potential copied segments. The segment-level metric assesses the amount of correctly predicted frames in both videos, which indicates how accurate the copied segments are temporally localized within the video pairs [19]. We also emphasize the importance of using the video-level metric. Standard precision and recall are used to evaluate for identified copied pairs. The video-level F-score evaluates how well the models can determine the presence or absence of copied segments in the video pairs, providing a complementary perspective on model performance.
Implementation Details. We incorporate ViT-small [7,37] as our backbone for the feature extractor. The input resolution is 384, patch size is 16, the embedding dimension is 384 and the output feature are projected to a dimension of 256. Each image of the DISC dataset [11] is used as query to retrieve the most similar 48 images from the dataset to form a mini-batch to train the model, with Adam optimizer and constant learning rate 1e-5 for 240K iterations. The detection model used for localization is YOLOX-s [13], which is trained using SGD with momentum 0.9, batch size of 64, initial learning rate of 0.01 and weight decay of 0.0005. More details can be found in our codes[1].

4.2 Comparison with Previous Methods

We compare our method with previous well-performing video copy localization methods on the VCSL and VCDB datasets in Table 1. The results of TransVCL are from the original paper [18]. The results of other comparing methods are reproduced using the code provided by the VCSL benchmark [19]. Previous works put more emphasis on the segment-level results [19], while we also report their video-level results. Note that previous methods perform differently under these two metrics. Some methods, such as TransVCL and SPD, are effective at localizing the copied segments with precise boundaries, resulting in good segment-level performance. On the other hand, methods such as HV and DP may lack the precision to pinpoint the exact location but are able to identify challenging copied segments in a video pair, leading to good video-level performance. This variation in performance demonstrates the importance of using both metrics to evaluate the capabilities of copy localization methods.

As shown in Table 1, our self-supervised video copy localization model (Ours-ssl), which is trained without using any human-annotated data, consistently outperforms previous methods in both segment-level and video-level F-scores. The performance can be further improved by fine-tuning the detector of Ours-ssl using the manually labeled VCSL training set, achieving segment and video level F-scores of 71.56% and 96.46%, respectively, which surpass the previous best models by a large margin. More analysis can be found in Section 4.4.

[1] https://github.com/eccv1818rtr/ECCV1818

Table 1: Video copy localization performance comparison on the VCSL and VCDB datasets. Our self-supervised model (Our-ssl) is trained without any human-annotated data. Our-ft model is obtained by fine-tuning the detector on the VCSL training set.

Dataset	Method	Segment-level			Video-level		
		Recall	Precision	**F-score**↑	Recall	Precision	**F-score**↑
VCSL	HV [9]	86.94	36.82	51.73	89.08	98.98	93.77
	TN [34]	62.49	66.50	64.43	82.88	99.95	90.62
	DP [5]	49.56	60.63	54.53	85.98	99.43	92.21
	DTW [3]	45.10	56.67	50.23	68.12	99.94	81.02
	SPD [22]	71.47	56.27	62.97	86.15	81.90	83.97
	TransVCL [18]	65.59	67.46	66.51	83.34	97.97	90.06
	Ours-ssl	69.51	68.17	**68.83**	91.35	99.87	**95.42**
	Ours-ft	75.76	67.81	**71.56**	93.93	99.14	**96.46**
VCDB	HV [9]	89.23	58.70	70.81	85.99	100.00	92.47
	TN [34]	73.01	77.54	75.21	73.58	100.00	84.78
	DP [5]	63.90	73.52	68.37	72.35	100.00	83.95
	DTW [3]	61.78	72.26	66.61	62.77	100.00	77.13
	SPD [22]	78.68	74.18	76.36	83.39	100.00	90.94
	TransVCL [18]	76.69	74.09	75.37	78.47	100.00	87.94
	Ours-ssl	78.98	75.61	**77.26**	87.46	100.00	**93.31**
	Ours-ft	80.74	76.91	**78.78**	88.89	100.00	**94.12**

*The VCDB dataset has no negative pairs, therefore the video-level precisions are 100%.

Table 2: The effectiveness of the Transitivity Property for generating copied video pairs. Performance is evaluated on the VCSL dataset, using the feature from previous methods (eff256d) as well as our Regional Token Representation (RTR).

Feature	Training Data	Segment-level			Video-level		
		Recall	Precision	**F-score**↑	Recall	Precision	**F-score**↑
eff256d	ssl-vanilla	69.39	61.42	65.16	89.36	99.26	94.05
	ssl-transitivity	68.04	66.78	67.40	88.96	99.90	94.12
RTR	ssl-vanilla	71.15	63.24	66.96	90.87	99.69	95.08
	ssl-transitivity	69.51	68.17	68.83	91.35	99.87	95.42

4.3 Ablation Study

We demonstrate the effectiveness of the Transitivity Property used to generate copied video pairs in Table 2. The models trained with video pairs generated by the Transitivity Propagation achieve better results than those trained with the vanilla strategy. The performance gain is greater at the segment level than at the video level. The Transitivity-based generation introduces a greater diversity of appearances in the similarity map, which is more beneficial for localizing precise segment boundaries in testing scenarios. As shown in Table 2, our self-supervised trained model outperforms all previous methods when utilizing the same feature, eff256d [18,19,41]. Furthermore, with the adoption of the Regional Token Representation, our model achieves even better results.

Table 3: Model performance for various configurations with differing numbers of tokens. The 1-token configuration corresponds to the vanilla ViT model, where the representation of the CLS token is used for computing similarity. The 2-token configuration denotes our model, which includes 1 CLS token and 1 Regional Token. Additionally, we add 1 and 2 tokens to build the 3-token and 4-token models, respectively.

	Segment-level			Video-level		
Number of Tokens	Recall	Precision	**F-score↑**	Recall	Precision	**F-score↑**
1 (1cls, ViT)	67.98	67.93	67.96	89.93	99.80	94.61
2 (1cls + 1r, ours)	69.51	68.17	68.83	91.35	99.87	95.42
3 (1cls + 1r + 1t)	69.76	68.13	68.93	91.18	99.90	95.34
4 (1cls + 1r + 2t)	68.92	68.27	68.59	90.86	99.82	95.13

Table 3 presents an analysis of how model performance varies with the use of different numbers of tokens on the VCSL dataset. We perform a similar analysis on the DISC dataset, following the standard metrics for evaluating image copy detection [11], as shown in Table 4. Our model, which contains 2 tokens, achieves better results than the vanilla ViT model on both the VCSL and DISC datasets. This demonstrates the effectiveness of the additional Regional Token employed in our model. The 3-token and 4-token models are built by adding more tokens and using stage-wise training based on our model. When adding more tokens, there is a slight increase in performance on the DISC dataset, but no significant performance gain is observed on the VCSL dataset. This suggests that more tokens may be beneficial for more complex cases of copy editing, but a single Regional Token is currently sufficient for the VCSL dataset. The plateau in performance with two tokens on the VCSL dataset could be attributed to the scarcity of video data containing multiple picture-in-picture local regions.

Table 4: Performance comparison of various configurations on the DISC dataset.

Number of Tokens	μAP	R@P90
1 (1cls, ViT)	55.39	40.81
2 (1cls + 1r, ours)	58.66	44.96
3 (1cls + 1r + 1t)	59.38	45.31
4 (1cls + 1r + 2t)	59.40	45.51

4.4 Comparison with Supervised Training

We compare our self-supervised model to its counterpart that has the same architecture but is trained in a supervised manner using human-annotated data. As illustrated in Table 5, our self-supervised model (Ours-ssl) achieves a higher video-level F-score than the supervised model (Ours-sup), while it has a lower F-score at the segment-level. Specifically, the segment-level performance is comparable to the Ours-sup model across regular topic categories, but Ours-ssl model underperforms in the "kichiku" category, as detailed in the Supplementary Material. This can be attributed to the nature of "kichiku" videos, which remix or mash up content from various sources and often contain repetitive and rapid temporal edits, which are challenging for an algorithm to mimic effectively. Example similarity maps are shown in Case C of Fig. 3 (C). The difference in

Table 5: The comparison of our self-supervised model (Ours-ssl) with its supervised counterpart (Ours-sup), where the detector is trained using human-annotated data from the VCSL dataset. We also provide the performance where Ours-ssl model is further fine-tuned using different percentages of the VCSL training set. Performance is evaluated on the VCSL testing data, utilizing the eff256d and our RTR features.

Feature	Method	Segment-level			Video-level		
		Recall	Precision	**F-score↑**	Recall	Precision	**F-score↑**
eff256d	*Supervised*						
	TransVCL [18]	65.59	67.46	66.51	83.34	97.97	90.06
	TransVCL-imp	69.63	67.90	68.76	87.84	99.67	93.38
	Ours-sup	71.85	66.77	69.22	88.25	99.11	93.37
	Self-supervised						
	Ours-ssl	68.04	66.78	67.40	88.96	99.90	94.12
	Self-supervised & fine-tuned						
	Ours-ft-1%	71.98	66.44	69.10	90.31	99.65	94.75
	Ours-ft-5%	73.33	66.57	69.78	89.72	99.32	94.28
	Ours-ft-20%	73.15	68.33	70.66	90.71	99.95	95.11
	Ours-ft	73.46	68.19	**70.73**	90.94	99.80	**95.17**
RTR	*Supervised*						
	Ours-sup	74.52	65.49	69.71	91.38	98.77	94.93
	Self-supervised						
	Ours-ssl	69.51	68.17	68.83	91.35	99.87	95.42
	Self-supervised & fine-tuned						
	Ours-ft-1%	70.21	69.58	69.90	90.51	99.89	94.97
	Ours-ft-5%	74.77	67.87	71.15	92.16	99.43	95.66
	Ours-ft-20%	75.54	67.43	71.26	92.36	99.64	95.86
	Ours-ft	75.76	67.81	**71.56**	93.93	99.14	**96.46**

performance of the models suggests a complementary property between the generated and human-annotated copy pairs.

The supervised model (Our-sup) shares the same backbone detector with the baseline model of TransVCL [18] but achieves better results. This improvement is attributed to our substitution of the dual-softmax similarity map used in [18] with a cosine similarity map. When using the same setting, the performance of TransVCL is also improved, as indicated by TransVCL-imp in Table 5. However, in this configuration, the added value of the Transformer layers incorporated in TransVCL appears to diminish.

We fine-tune the detector of our self-supervised model with different percentages of the VCSL training set. With only 1% of the manually-labeled data, our model achieves similar or slightly better results than the fully-supervised model (Our-sup) which is trained using the entire training set. When fine-tuned on the entire training set, our model (Ours-ft) achieves even better results, attain-

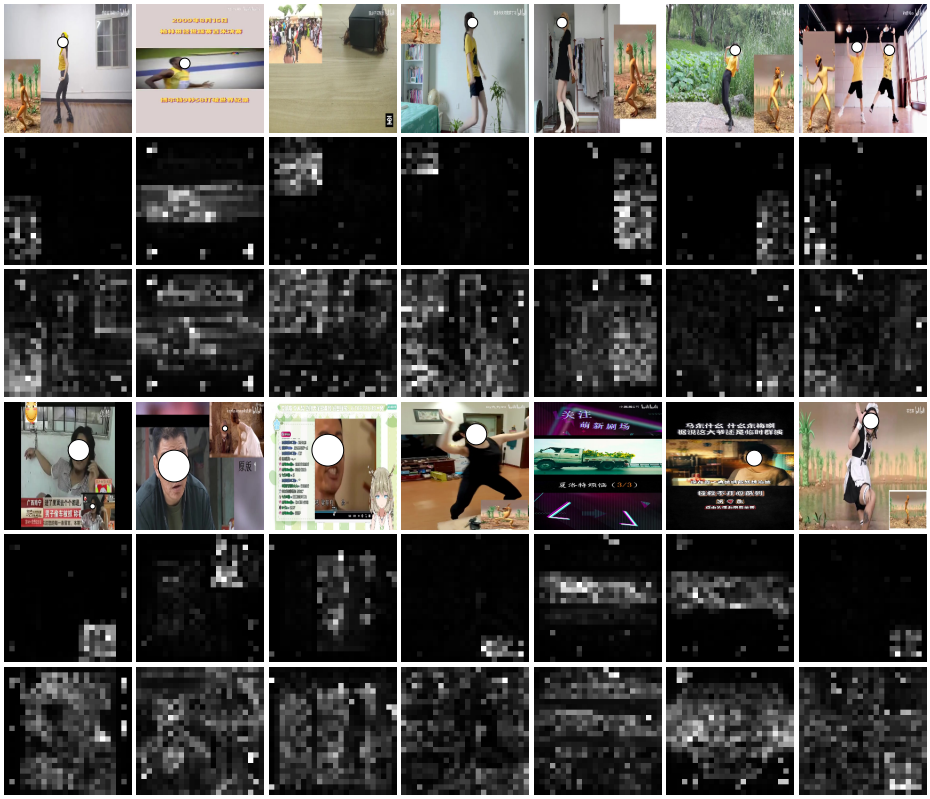

Fig. 4: Attention maps of our model. Row 1 and 4: the original frames; Row 2 and 5: the attention maps for the Regional Token; Row 3 and 6: the attention maps for the CLS token. White circles are drawn over human faces to protect identities.

ing segment-level and video-level F-scores of 71.56% and 96.46%, respectively, surpassing previous methods by a large margin, as detailed in Table 1.

4.5 Visualization

Figure 4 shows the self-attention maps for the Regional Token and the CLS token of the last layer in our feature extractor. Our input has a resolution of 384, which results in a sequence of 578 tokens, including the Regional Token and the CLS token. We take the attention weights for the patch tokens when the Regional Token and CLS token serve as the queries. These weights are then averaged across all heads and scaled to the range of [0, 255] to produce the visualization. It can be observed that the attention of our Regional Token is primarily focused on the local regions, whereas the CLS token tends to concentrate more on global information. This demonstrates the effectiveness of our Regional Token in capturing information from picture-in-picture areas, which is crucial for detecting the copied content.

Table 6: Comparison with recent competition winning solutions on the VCSL dataset.

Dataset	Method	Segment-level			Video-level		
		Recall	Precision	**F-score**↑	Recall	Precision	**F-score**↑
VCSL	MultiCrop [39]	74.38	69.93	72.09	93.10	99.79	96.33
	ImConcat [21]	57.54	67.19	62.00	91.60	97.83	94.61
	SAM [28]	65.87	67.51	66.68	91.60	96.40	93.94
	Ours-ft	75.76	67.81	71.56	93.93	99.14	96.46

4.6 Comparison with Competition Winners

We adapt the winning solutions [21, 28, 39] from recent image and video competitions [11, 30] and evaluate them on the VCSL dataset. The approach in [39] employs a multiple crops strategy, we implement a 13-crop version and extract features for each crop for similarity measure. Two images are concatenated as input for a Vision Transformer (ViT) in [21]. We apply this method by feeding the corresponding frame pair to the model to compute each pixel of the similarity map for the video pair. The Similarity Alignment Model (SAM) proposed in [28] are also evaluated. These methods are not directly comparable to our model, as their excessive processing requires computational costs that are at least an order of magnitude larger (More details in Supplementary Material). As shown in Table 6, our model, while being much simpler and more time-efficient, achieves similar or better performance than these methods.

5 Conclusion and Discussion

In this work, we propose a self-supervised approach for video copy localization, where our feature extractor and temporal localization model are both trained in a self-supervised manner. By adding a Regional Token to the ViT model and utilizing an asymmetric training procedure, our feature extractor effectively learns regional representations and improves performance. We introduce a novel strategy that leverages the Transitivity Property to generate copied video pairs, which facilitates the training of the temporal localization model. Without the use of any human-annotated data, our model achieves state-of-the-art performance on the VCSL and VCDB datasets.

Ethical considerations. We use public datasets DISC [11], VCSL [19], and VCDB [23] in our experiments. To protect identities, human faces in the result figures have been masked. Copy detection is an adversarial task, and there is a risk that publishing research on this problem may offer insights that enable users to circumvent moderation systems. However, we believe that research in this direction provides a net benefit to society, as also discussed in [19, 31].

Future work. Our feature extractor is limited by being trained with only image input, without considering the relationships between frames. It remains unclear whether and how temporal information in videos is useful for the copy localization task. This represents an interesting direction for future work.

References

1. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. In: ICLR (2022)
2. Baraldi, L., Douze, M., Cucchiara, R., Jégou, H.: Lamv: Learning to align and match videos with kernelized temporal layers. In: CVPR. pp. 7804–7813 (2018)
3. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD. pp. 359–370 (1994)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607. PMLR (2020)
5. Chou, C.L., Chen, H.T., Lee, S.Y.: Pattern-based near-duplicate video retrieval and localization on web-scale videos. TMM **17**(3), 382–395 (2015)
6. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV. pp. 1422–1430 (2015)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Douze, M., Jégou, H., Sandhawalia, H., Amsaleg, L., Schmid, C.: Evaluation of gist descriptors for web-scale image search. In: ACM International Conference on Image and Video Retrieval. pp. 1–8 (2009)
9. Douze, M., Jégou, H., Schmid, C., Pérez, P.: Compact video description for copy detection with precise temporal alignment. In: ECCV. pp. 522–535. Springer (2010)
10. Douze, M., Revaud, J., Verbeek, J., Jégou, H., Schmid, C.: Circulant temporal encoding for video retrieval and temporal alignment. IJCV **119**, 291–306 (2016)
11. Douze, M., Tolias, G., Pizzi, E., Papakipos, Z., Chanussot, L., Radenovic, F., Jenicek, T., Maximov, M., Leal-Taixé, L., Elezi, I., et al.: The 2021 image similarity dataset and challenge. arXiv preprint arXiv:2106.09672 (2021)
12. Feichtenhofer, C., Li, Y., He, K., et al.: Masked autoencoders as spatiotemporal learners. NeurIPS **35**, 35946–35958 (2022)
13. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
14. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. NeurIPS **33**, 21271–21284 (2020)
15. Han, Z., He, X., Tang, M., Lv, Y.: Video similarity and alignment learning on partial video copy detection. In: ACMMM. pp. 4165–4173 (2021)
16. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR. pp. 16000–16009 (2022)
17. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR. pp. 9729–9738 (2020)
18. He, S., He, Y., Lu, M., Jiang, C., Yang, X., Qian, F., Zhang, X., Yang, L., Zhang, J.: Transvcl: Attention-enhanced video copy localization network with flexible supervision. In: AAAI. vol. 37, pp. 799–807 (2023)
19. He, S., Yang, X., Jiang, C., Liang, G., Zhang, W., Pan, T., Wang, Q., Xu, F., Li, C., Liu, J., et al.: A large-scale comprehensive dataset and copy-overlap aware evaluation protocol for segment-level video copy detection. In: CVPR. pp. 21086–21095 (2022)

20. He, X., Pan, Y., Tang, M., Lv, Y., Peng, Y.: Learn from unlabeled videos for near-duplicate video retrieval. In: ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1002–1011 (2022)
21. Jeon, S.: 2nd place solution to facebook ai image similarity challenge matching track. arXiv e-prints pp. arXiv–2111 (2021)
22. Jiang, C., Huang, K., He, S., Yang, X., Zhang, W., Zhang, X., Cheng, Y., Yang, L., Wang, Q., Xu, F., et al.: Learning segment similarity and alignment in large-scale content based video retrieval. In: ACMMM. pp. 1618–1626 (2021)
23. Jiang, Y.G., Jiang, Y., Wang, J.: Vcdb: a large-scale database for partial copy detection in videos. In: ECCV. pp. 357–371. Springer (2014)
24. Kim, C.: Content-based image copy detection. Signal Processing: Image Communication **18**(3), 169–184 (2003)
25. Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., Kompatsiaris, Y.: Visil: Fine-grained spatio-temporal video similarity learning. In: ICCV (2020)
26. Kordopatis-Zilos, G., Tolias, G., Tzelepis, C., Kompatsiaris, I., Patras, I., Papadopoulos, S.: Self-supervised video similarity learning. In: CVPRW. pp. 4755–4765 (2023)
27. Kordopatis-Zilos, G., Tzelepis, C., Papadopoulos, S., Kompatsiaris, I., Patras, I.: Dns: Distill-and-select for efficient and accurate video indexing and retrieval. IJCV **130**(10), 2385–2407 (2022)
28. Liu, Z., Ma, F., Wang, T., Rao, F.: A similarity alignment model for video copy segment matching. arXiv preprint arXiv:2305.15679 (2023)
29. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: ECCV. pp. 527–544. Springer (2016)
30. Pizzi, E., Kordopatis-Zilos, G., Patel, H., Postelnicu, G., Ravindra, S.N., Gupta, A., Papadopoulos, S., Tolias, G., Douze, M.: The 2023 video similarity dataset and challenge. arXiv preprint arXiv:2306.09489 (2023)
31. Pizzi, E., Roy, S.D., Ravindra, S.N., Goyal, P., Douze, M.: A self-supervised descriptor for image copy detection. In: CVPR. pp. 14532–14542 (2022)
32. Qian, R., Meng, T., Gong, B., Yang, M.H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal contrastive video representation learning. In: CVPR. pp. 6964–6974 (2021)
33. Recasens, A., Luc, P., Alayrac, J.B., Wang, L., Strub, F., Tallec, C., Malinowski, M., Pătrăucean, V., Altché, F., Valko, M., et al.: Broaden your views for self-supervised video learning. In: ICCV. pp. 1255–1265 (2021)
34. Tan, H.K., Ngo, C.W., Hong, R., Chua, T.S.: Scalable detection of partial near-duplicate videos by visual-temporal consistency. In: ACMMM. pp. 145–154 (2009)
35. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of cnn activations. arXiv preprint arXiv:1511.05879 (2015)
36. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. NeurIPS **35**, 10078–10093 (2022)
37. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML. pp. 10347–10357. PMLR (2021)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017)
39. Wang, W., Sun, Y., Zhang, W., Yang, Y.: D2lv: A data-driven and local-verification approach for image copy detection. arXiv preprint arXiv:2111.07090 (2021)
40. Wei, D., Lim, J.J., Zisserman, A., Freeman, W.T.: Learning and using the arrow of time. In: CVPR. pp. 8052–8060 (2018)

41. Yokoo, S.: Contrastive learning with large memory bank and negative embedding subtraction for accurate copy detection. arXiv preprint arXiv:2112.04323 (2021)
42. Zhang, J., Zhu, W., Li, B., Hu, W., Yang, J.: Image copy detection based on convolutional neural networks. In: Chinese Conference on Pattern Recognition. pp. 111–121. Springer (2016)
43. Zhou, W., Lu, Y., Li, H., Song, Y., Tian, Q.: Spatial coding for large scale partial-duplicate web image search. In: ACMMM. pp. 511–520 (2010)