Unsupervised Open-Vocabulary Object Localization in Videos

Ke Fan[1,*], Zechen Bai[2,*], Tianjun Xiao[2], Dominik Zietlow[2], Max Horn[2], Zixu Zhao[2],
Carl-Johann Simon-Gabriel[2], Mike Zheng Shou[3], Francesco Locatello[2],
Bernt Schiele[2], Thomas Brox[2], Zheng Zhang[2,†], Yanwei Fu[1,†], Tong He[2]
[1]Fudan University [2]Amazon Web Services [3]National University of Singapore
kfan21@m.fudan.edu.cn, {baizeche, tianjux, zhaozixu, zhaz, htong}@amazon.com
{zietld, hornmax, cjsg, locatelf, bschiel, brox}@amazon.de
mikeshou@nus.edu.sg, yanweifu@fudan.edu.cn

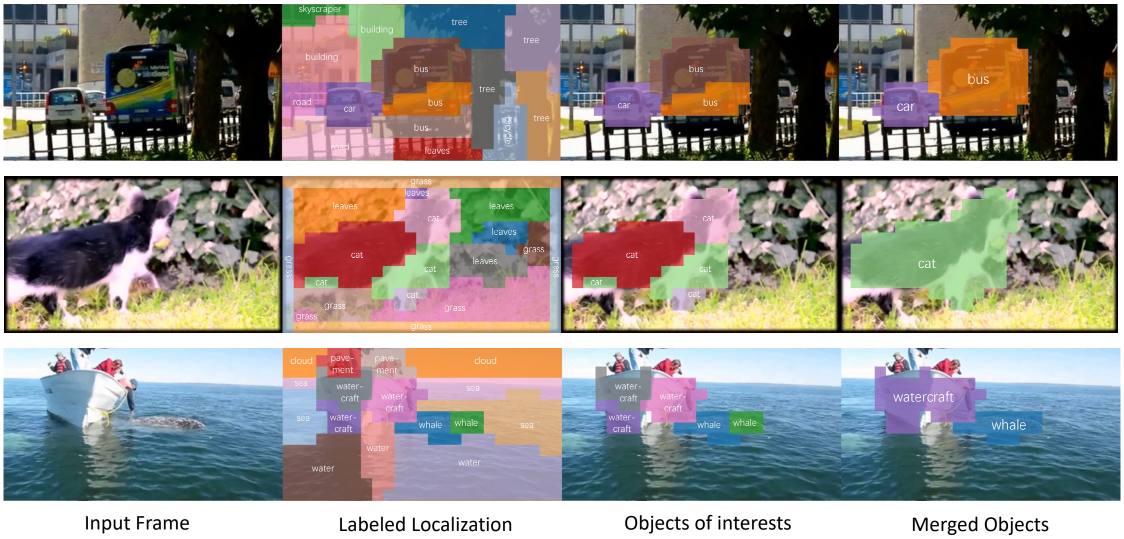


Figure 1. We propose an unsupervised approach to localize and name objects in real-world videos. The approach uses slot attention in feature space to localize tubes (second column), assigns text to the slot features via a CLIP model that was modified to allow local feature alignment (third column), and finally merges slots that overlap in text space (last column).

Abstract

In this paper, we show that recent advances in video representation learning and pre-trained vision-language models allow for substantial improvements in self-supervised video object localization. We propose a method that first localizes objects in videos via a slot attention approach and then assigns text to the obtained slots. The latter is achieved by an unsupervised way to read localized semantic information from the pre-trained CLIP model. The resulting video object localization is entirely unsupervised apart from the implicit annotation contained in CLIP, and it is effectively the first unsupervised approach that yields good results on regular video benchmarks.

1. Introduction

Deep learning has demonstrated remarkable performance in object-detection and -recognition, both on images and videos. Most models are trained via supervised learning, which requires expensive manual annotation of training data. Such approaches also run the risk of overfitting to specific distributional characteristics of the training data, and may not generalize well beyond that. Self-supervised learning methods are a promising approach to overcome this problem. They can learn from vast amounts of unlabeled visual data and unlock the full potential of deep learning, for example in video understanding.

Recently, a lot of progress was made in training segmentation models without supervision. Caron et al. [2]

[*] Equal contribution. Ke Fan is the first intern author, Zechen Bai is the first FTE author, they contributed equally. Work down during Ke Fan's internship in AWS Shanghai AI Lab

[†] Corresponding authors. Yanwei Fu and Ke Fan are with the School of Data Science, Fudan University.

demonstrated that vision transformers, combined with self-supervised learning, can create a feature space in which patterns of the same object class cluster. This feature space was used in a series of papers to learn unsupervised segmentation models [14, 26, 19]. In another line of research, Locatello et al. [13] proposed an approach, where so-called *slots* localize individual objects in the scene and represent their visual patterns and properties. This type of approach belongs to the field of *object-centric learning (OCL).* So far, much of the OCL literature has focused on static images. Only recently it became possible to obtain good results not only on synthetic but also on real-world datasets [19]. Few papers have tried to apply a slot-based OCL approach to video data [11, 6]. These works scale to real-world data by adding some form of weak annotation.

In contrast, this paper proposes an OCL pipeline for *real-world* video data with two main goals: (1) partitioning the videos into spatially, temporally, and semantically meaningful groups *without using any labeled training data*; and (2) labeling those groups using an off-the-shelf vision-language model, such as CLIP [17]. Since CLIP was trained to align text with global image features, it is initially unable to align text with local features from individual objects. However, CLIP can be fine-tuned on image-text data to make it align the local patterns to text [15]. In this paper, we show that paired image-text data is not necessary to allow for such alignment. We propose a self-supervised objective to fine-tune the last layer of the CLIP vision transformer, which trains on image data only.

Overview. The overall framework is composed of three parts, as depicted in Figure 2. The individual processing stages are visualized in Figure 1. The first part yields bottom-up object-centric video representations (slots) that localize the regions in the video with a spatio-temporal grouping model (Section 3). The second part adapts the off-the-shelf image-text alignment model CLIP [17] to assign a text to the video slots (Section 4). Finally, a merging procedure leverages overlapping information from the text and the image to improve both localization and labeling (Section 5). Our contributions are twofold:

- We provide the first approach that localizes objects with spatio-temporal consistency in real-world videos, without labelled training data.
- We assign text labels to video slots using a pre-trained CLIP model without additional supervised fine-tuning.

2. Related Work

Object-centric localization. Our research contributes to the field of object-centric learning, which involves extracting individual objects from visual input using neural networks with inductive biases. Some approaches, such as using an information bottleneck with pixel-level reconstruction [21, 13, 24] or examining perceptual similarity through self-supervised representation pretraining [7, 4], have been previously explored. Our contribution is introducing temporal coherence into the mix, which improves object emergence and enhances the objectness signal by leveraging the Common Fate principle. Furthermore, there have been recent advances in object-centric learning for video applications [11, 6, 21, 1]. These approaches improve upon previous image-level methods by incorporating temporal modules into the network architecture. However, they either require additional annotation signals in the form of the first frame and optical flow, or they rely on complex modules to enable temporal tracking or matching of object slots in neighboring frames. In comparison, our method does not require any additional signals or specifically designed network modules, yet still achieves competitive performance on real-world videos.

CLIP for low-level vision tasks. Recent advances in vision-language models have been driven by large-scale Contrastive Language-Image Pre-Training (CLIP) model [17]. While CLIP has shown promising results in low-level vision tasks, its training setup has been criticized for lacking localization ability [15, 12, 5, 25]. Some solutions involve using gradient activation maps [20] or extra annotation to finetune CLIP [25]. Our contribution is decoupling the tasks of object-centric learning and vision-language, allowing us to finetune CLIP with only a moderate *unlabeled* dataset for semantic feature extraction, without the need for additional annotations. This approach stands in contrast to previous works that require additional training annotations for both localization and semantic labeling using a single vision-language module [25].

3. Video slot learning

Problem setup. Given a video $V \in \mathbb{R}^{T\times H\times W\times 3}$, the task is to extract K video slots $S_{\text{video}} \in \mathbb{R}^{K\times D_{\text{slot}}}$ that bind to object features in the video, as well as a video alpha mask $\alpha \in \mathbb{R}^{K\times T\times H\times W}$ used to segment the video into K tubes.

Self-supervised video encoder. For a pre-defined $p \times p \times 1$ patch size, we tokenize the video V and flatten the spatio-temporal dimension to a sequence of tokens $\text{tn}_V = \text{Tokenize}(V) \in \mathbb{R}^{T\times H'\times W'\times D}$, where $H' = \lceil \frac{H}{p} \rceil$ and $W' = \lceil \frac{W}{p} \rceil$. After adding spatio-temporal positional encoding, the tokens are processed by a video encoder E_{V} as D-dimensional video token features, $F = \mathrm{E}_{\mathrm{V}}(V) \in \mathbb{R}^{T\times H'\times W'\times D}$. The video encoder consists of a stack of transformer blocks, an architecture that has been shown effective for self-supervised vision representation learning without annotation [2, 22, 8].

Learning video slots. We generalize DINOSAUR [19], a state-of-the-art object-centric learning method for *static* images, to videos. DINOSAUR passes the tokens of an im-

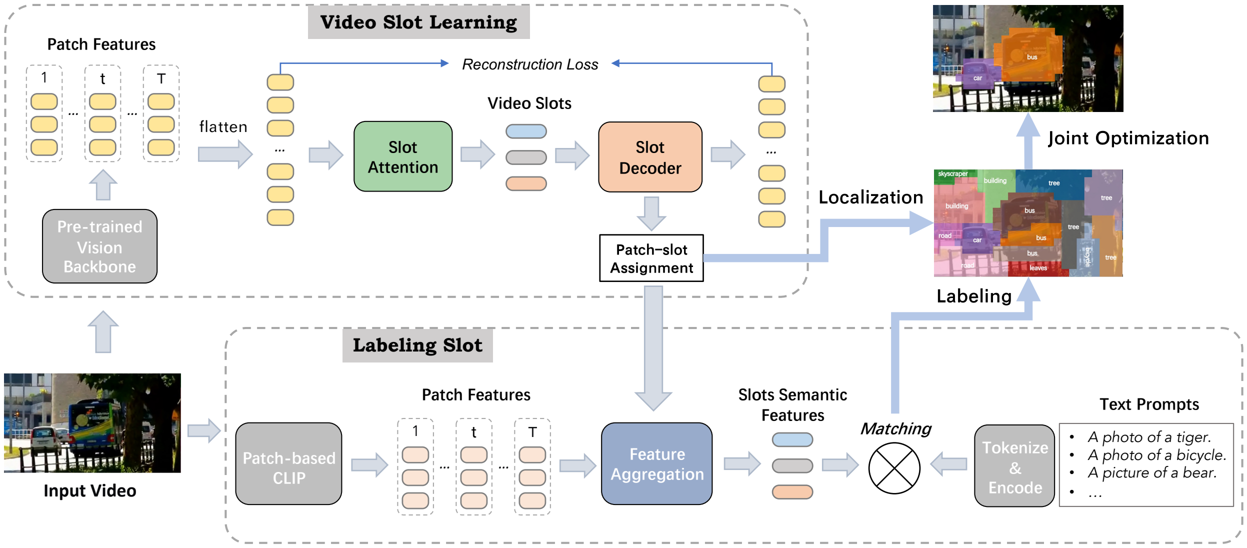


Figure 2. **Proposed framework.** Given an input video, we first localize objects by slot attention with a video encoder pretrained with self-supervision. Next, we extract semantic features for each slot by a patch-based CLIP finetuned from its vanilla version. Then, slots are named by matching slot semantic features to text features from a curated list of text prompts. Finally, the named slots are optimized to alleviate over-segmentation caused by part-whole hierarchies.

age, *e.g.*, a frame at time step t through a pre-trained image encoder $\mathrm{E_I}$ and obtains D-dimensional image features $f_t = \mathrm{E_I}(\mathrm{tn}_{V_t}) \in \mathbb{R}^{H'\times W'\times D}$. It then flattens the spatial dimensions of f_t and applies slot-attention [13] as follows:

$$S_{\text{img},t} = \mathrm{SlotAttention}(f_t.\text{reshape}(H'W', D)).$$

$S_{\text{img},t}$ effectively represents image slots, each corresponding to a group of pixels that are supposed to cover an object.

To use DINOSAUR on videos, a straightforward approach would be to run it separately on each frame of the video. However, this require to run slot-attention T times and results in poor temporal consistency among slots in multiple frames. Instead, we run slot-attention only once on the entire spatio-temporal features $F \in \mathbb{R}^{T\times H'\times W'\times D}$ of the video. Specifically, we flatten the spatial and temporal dimensions and extract video slots for all frames as follows:

$$S_{\text{video}} = \mathrm{SlotAttention}(F.\text{reshape}(TH'W', D)) \in \mathbb{R}^{K\times D_{\text{slot}}}.$$

S_{video} are video slots that capture video features. Since patches from a same object tend to have similar features across subsequent frames, the slot attention mechanism tends to group them into a same slot, which ensures temporal consistency *by design*.

Slot decoder and training loss. With S_{video}, each slot $S_i \in \mathbb{R}^{D_{\text{slot}}}$ is first boardcasted to a 3D volume $V_i \in \mathbb{R}^{T\times H\times W\times D_{\text{slot}}}$ and added to a spatio-temporal positional encoding pe, then a 2-layer Multi-Layer Perceptron (MLP) is applied on each position individually:

$$y_i, \alpha_i = \mathrm{MLP}(V_i + pe)$$

where we have $\mathbf{y} = [y_1, \ldots, y_K]$ with $y_i \in \mathbb{R}^{T\times H'\times W'\times D}$ as the feature reconstruction, and $\alpha = [\alpha_1, \ldots, \alpha_K]$ with $\alpha_i \in \mathbb{R}^{T\times H'\times W'}$ as the alpha mask for the patch-to-slot assignment weight. Specifically, let $\alpha_{\cdot,j} = [\alpha_{1,j}, \ldots, \alpha_{K,j}]$ be the alpha mask weight for the j-th patch, then $\sum_{i=1}^{K} \alpha_{i,j} = 1$. We say that the j-th patch belongs to the i^*-th slot if $i^* = \arg\max_i \alpha_{i,j}$. Similar to previous work [21, 19], during training, we use the following reconstruction loss of video features F based on the slot-attention output $\mathbf{y}$ and α:

$$L = \mathrm{MSE}(\mathbf{y} \cdot \alpha, F).$$

4. Labeling video slots

Problem Setup. Given a video $V \in \mathbb{R}^{T\times H\times W\times 3}$, its alpha mask $\alpha \in \mathbb{R}^{K\times T\times H'\times W'}$ and a list of N potential labels and corresponding features $f_{\text{text}} \in \mathbb{R}^{N\times D_{\text{sem}}}$, the task is to come up with a slot-to-label assignment matrix $A \in \mathbb{R}^{K\times N}$

A naive baseline with CLIP. CLIP [17] is a vision-language model pre-trained on an internet-scale dataset. For each image-text pair, CLIP summarizes the text and vision features in tokens CLS_t and CLS_v and aligns them using contrastive learning. The tokens CLS are constructed using an N-stack transformer. Since CLIP can be used to label almost any full image, a straightforward approach for slot labeling would be to wrap the video slots into bounding boxes, crop the regions and send each of them to CLIP. However, this method suffers from two problems. First, this requires multiple runs of the CLIP visual encoder for one input image. Second, when a slot is non-convex in shape, its corresponding bounding box would inevitably include content that does not belong to the slot and eventually bias the features from CLIP. In the following, we describe a more efficient and unbiased approach to slot labeling which requires minor modifications on the pre-trained CLIP model but no additional labels.

Efficient slot labeling with semantic features. Assume there is an image semantic encoder E_{sem} that operates on the same patch size p as in Section 3. We encode the t-th frame I_t into a set of semantic patch features $\{P_t\} = E_{\text{sem}}(I_t) \in \mathbb{R}^{H' \times W' \times D_{\text{sem}}}$. For each video slot, we average the patch features in this slot as the slot semantic feature $\{f_{\text{slot}}\} \in \mathbb{R}^{K \times D_{\text{sem}}}$. Given a list of potential semantics and corresponding features $\{f_{\text{text}}\} \in \mathbb{R}^{N \times D_{\text{sem}}}$, we compute the cosine similarity between f_{slot} and f_{text}. Let A be the matrix for slot-text similarity: $A_{ij} = \cos(f_{\text{slot},i}, f_{\text{text},j})$. For each slot, we find the index of the text with maximum feature cosine similarity $j^* = \arg\max_j A_{ij}$ and label the slot accordingly.

Local semantics from CLIP. Although the output of CLIP's visual encoder contains both the CLS_v token as well as image patch tokens, the contrastive loss used during pre-training only enforces alignment between the text token CLS_t with CLS_v, not with the other visual tokens. And in practice, those other patch tokens do not align with the text features CLS_t, as observed by [15, 12], and confirmed by our own initial experiments. Therefore, the visual encoder from a vanilla CLIP cannot be used without further modification as the semantic encoder E_{sem}. To fix this issue, [15, 12] adapt the CLIP visual encoder to local semantic features by fine-tuning it on additional vision data with image-text pairing, for example segmentation masks [12].

Patch-based CLIP. In contrast, we adapt the pre-trained CLIP visual encoder to a *patch-based CLIP* visual encoder using only *unlabeled images*, as illustrated in Figure 3. The idea is based on this observation: the semantic-informative token CLS_v is the output from a ViT's last layer, thus we can train a new function that re-projects the information from the last layer's input to new patch features.

Formally, given an input image I and the pre-trained CLIP visual encoder E^v_{CLIP}, we consider both its output token $CLS_v \in \mathbb{R}^{D_{\text{sem}}}$ as well as the patch tokens $P' \in \mathbb{R}^{M \times D_{\text{sem}}}$ of the ViT that were used in the penultimate layer to construct CLS_v. We replace the last layer by a new multi-head self-attention module E^v_{MHSA} and obtain new patch tokens $\hat{P} = \text{E}^v_{\text{MHSA}}(P')$.

This new module is trained with self-supervision, under the assumption that a pre-trained CLIP has its CLS_v close to the corresponding text token CLS_t. We use CLS_v as a proxy to the semantic representation space. We first compute a cross-attention with queries being CLS_v, while keys and values are both $\hat{P}$:

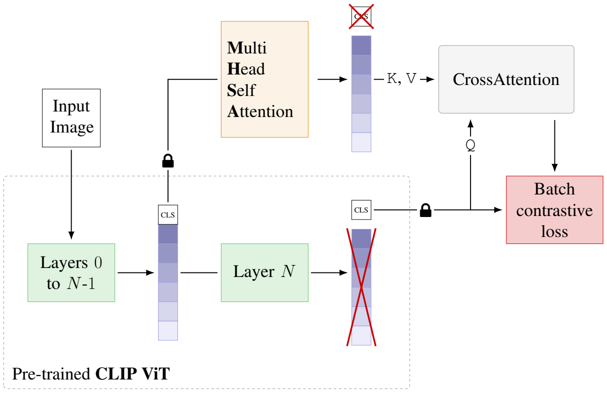


Figure 3. **Patch-based CLIP finetuning**. We replace the last ViT layer by a multi-head self-attention module to re-project semantic information to the new patch tokens. We then use cross-attention to encourage those patches that contain the main context to be similar to the CLS_v token. Maybe surprisingly, this suffices to get semantically meaningful patch features. Importantly, we do not use any labeled data during this fine-tuning step.

$$\hat{P}_v = \text{Softmax}(\hat{P} \cdot CLS_v^T) \cdot \hat{P}. \tag{1}$$

We then minimize the InfoNCE loss [16] between the aggregated feature $\hat{P}_v$ and CLS_v across data samples.

$$\phi_{ij} = \frac{\hat{P}_{v,i}}{|\hat{P}_{v,i}|} \cdot \frac{CLS_j}{|CLS_j|}, \tag{2}$$

$$\mathcal{L}_{\text{InfoNCE}} = \frac{1}{k} \sum_i^k \log\left(\frac{e^{\phi_{ii}}}{\sum_j^k e^{\phi_{ij}}}\right). \tag{3}$$

We use this scheme to adapt pre-trained CLIP patch features to the image-text joint space while training only on unlabeled image data. We do not use any visual annotation or image-text pairs. Intuitively, our modification of CLIP simply allows us to read out the information that was already contained in the CLIP model, but was not readily available.

5. Post-processing and joint optimization

The localization and text assignment procedures above already provide a good basis, but they leave room for improvements. In the following, we describe two post-processing steps that improve clustering on the text and the video data, respectively.

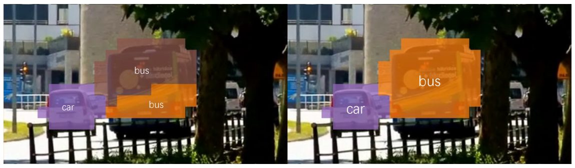
Optimizing labels with curated vocabulary. The similarity-based algorithm described from Section 4 requires a vocabulary of possible labels. In principle, the patch-based CLIP is capable to embed and recognize any word that appears in its large-scale training set. However, a too large vocabulary can lead to sub-optimal results due to synonyms and to less robust ranking performance against a too-long list of text features.

In practice, a good vocabulary should include both *target labels* and *common background labels*. Target labels are the dataset specific foreground object categories of interest. Common background labels are concepts that usually appear in the background and do not overlap with target labels.

Without prior knowledge, it is still possible that content of a slot may not correspond to any category, neither in the target, nor in common background labels. As a result, such a slot would have features that are not similar to any text features. We therefore leave slot i "unnamed" if $\cos(f_{\text{slot},i}, f_{\text{text},j}) < \lambda$ for all j, for some constant threshold λ. In practice, we find λ can be simply set to 0.

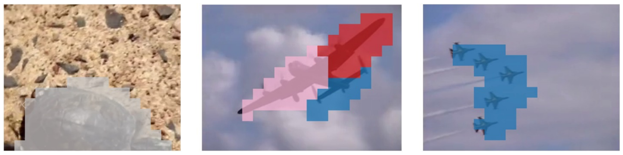
Improving part-whole with name-assisted localization. Using the slot labels (i.e., names), we can identify slots that cover only part of an object and merge them into a new slot with the same name.

Specifically, for each pair of slots S_i and S_j, we consider them as being part of the same object if they have the same label and are spatial neighbors in at least one frame. This merging process is repeated until there's no pair of slots to merge. Figure 4 illustrates this process. Although seemingly simple, this step is crucial to harness the part-whole issue. It builds both parts of our pipeline so far: the localization and the semantic label information. This concludes the description of our approach. The next section describes our evaluation pipeline.



Labeled Video Slots   Merged Video Slots

Figure 4. **Joint optimization**. The image on the left shows one frame of video slots with target names, the image on the right shows the result from merging. The two slots for the bus are merged thus better localizing the object, while they don't further merge with the slot for car since they share different semantics.



Single Object   Part of an Object   Group of Objects

Figure 5. **Three types of slots.** We show examples of a slot containing a single object, part of an object or a group of objects. For simplicity we only colorize slots overlapping with objects, and slots in one image are colored differently.

6. Datasets, implementations and baselines

Implementation details. (1) *Vision backbone.* We use the original VideoMAE module with two modifications: 1) we set each token to represent a $p \times p \times 1$ data patch instead of $p \times p \times 2$, and 2) we use 3D trigonometric function combination as the positional encoding. This modified VideoMAE is pre-trained on Something-Something-v2 [9] without supervision from human annotation. (2) *Patch-based CLIP* We train the patch-based CLIP model on ImageNet-1K [18] dataset without any class label. Specifically, we train a multi-head self-attention network to replace the last layer of the CLIP vision Transformer encoder.

Other baselines. For video slot learning, we compare against a number of OCL pipelines that output slots for each frame. Those baselines do not perform slot association across frames and do not produce video slots (i.e., K coherent tubes for a video). Nevertheless, their output can be sent into the patch-based CLIP to be named. (1) *Slot-Attention* [13] uses a convolutional neural network to encode an image and extracts K object-centric features using slot attention. (2) *SAVi* [11] is an extension of Slot Attention to videos that initializes the slots with some form of weak supervision, and then uses the slot from the previous frame to initialize the current frame. Since our method is completely unsupervised, for a fair comparison, we do not provide this weak annotation in our evaluation. (3) *DINOSAUR* [19] is a pixel grouping method that manages to group pixels on real-world data by freezing a pre-trained network as the backbone and reconstructing the extracted features. Out of the three baselines, only DINOSAUR uses a pre-trained backbone (self-supervised with DINO [2] on unlabeled ImageNet-1K).

Datasets. (1) *ImageNet-VID* (I-VID) is a common benchmark for video object detection which includes 30 object categories of the ImageNet DET dataset. (2) *YouTube-VIS* (Y-VIS) is a large-scale video instance segmentation dataset, which also includes object bounding boxes.

Metrics. To evaluate the localization performance of our method, we apply standard metrics in *object localization* tasks. Similar to previous studies [19, 23], we report *CorLoc*, which is the proportion of images where at least one object was correctly localized, and *DecRate*, which measures the percentage of objects that are correctly detected out of all the ground-truth objects. To further evaluate the overall labeling performance, we use the traditional object detection metric mean Average Precision (mAP), which examines the accuracy of both object localization and semantic classification.

Besides, to understand how well our framework learns video slots, we quantify whether a slot indeed captures a single object. Empirically, we observe that there are several other typical cases that deviate from that ideal behavior. To quantitatively analyze the challenge posed by part-whole hierarchies, we categorize slots that overlap with objects based on whether they contain a single object (SO), part of an object (PO) or a group of objects (GO). We label

Method	ImageNet-VID			Youtube-VIS		
	CorLoc	DecRate	mAP$_{50}$	CorLoc	DecRate	mAP$_{50}$
Slot-Attention	11.64	7.83	0.51^*	10.27	6.57	0.51^*
SAVi	16.23	10.55	1.23^*	11.10	7.18	0.90^*
DINOSAUR	48.04	39.58	7.34^*	44.80	34.63	8.15^*
Ours	**60.90**	**53.75**	**29.23**	**63.74**	**55.05**	**35.19**

Table 1. **Comparison with other object-centric methods on object localization and slot labeling.** Note that numbers with * superscript means that the method uses our slot labeling module when computing the data.

all those that do not overlap with any object or have only minor overlap with an object as background (BG). See Figure 5 for a qualitative explanation and Appendix for details. Please note that *the classification of these categories is only meant to give a qualitative and informative assessment of the distribution of errors*.

7. Experiments

The complete design space is large, it includes, for video slot learning, 1) image (e.g., DINO) versus video backbone (e.g., VideoMAE) and 2) spatio-temporal versus per-frame grouping, and for video slot labeling, 3) vanilla CLIP versus patch-aligned adaptation and 4) joint optimization using both text and image features. This paper does not attempt to exhaust this space. However, we will provide evidence that combining a video backbone with spatio-temporal grouping gives the best video slot learning ability, that CLIP adaptation is crucial, and that joint learning proves to be very effective. This conclusion is not at all unexpected. We now proceed to examine our (incomplete) evaluations with a set of questions.

Q1: How does our complete pipeline compare to the baselines? We present a quantitative evaluation on two datasets in Table 1. As pixel-reconstruction methods, slot-attention and SAVi perform poorly on both real-world video datasets across all metrics, confirming the ineffectiveness of the pixel-space reconstruction loss. DINOSAUR, which relies on feature reconstruction, performs better, particularly in terms of CorLoc and DecRate. However, its mAP score is only around 8, because it fails to align localization with semantics. In contrast, our model outperforms all other approaches across all metrics on both datasets, thanks to its ability to achieve spatio-temporal consistency and joint optimization with semantics.

Q2: Does the choice of vision backbone and grouping algorithm matter for slot learning? As described in Section 3, video slot learning depends on the choice of vision backbone and grouping algorithm. Table 2 demonstrates that a video-based backbone, such as VideoMAE, leads to superior performance in spatio-temporal slot learning across all metrics. Conversely, the spatial-only slot learning produces better results with an image-based backbone. Therefore, the combination of video backbone and grouping algorithm is crucial and must be selected accordingly.

Backbone	Video Slot	CorLoc	DecRate	mAP$_{50}$
DINO	×	62.12	54.79	28.75
	✓	59.84	52.98	28.40
VideoMAE	×	57.16	50.25	25.77
	✓	60.90	53.75	29.23

Table 2. **Ablations on vision backbones and video slots learning algorithm** on ImageNet-VID.

Although spatial DINO outperforms our model in terms of CorLoc and DecRate, we contend that these metrics pertain to image-based approaches. The higher mAP$_{50}$ score suggests that the temporal consistency ultimately improves the correct labeling and overall detection performance. To further illustrate the benefits of temporal consistency, we provide additional qualitative visualizations in Figure 6.

Q3: How important is CLIP-based joint optimization? To address the importance of image-text alignment, we conduct a quantitative comparison of joint optimization with and without image-text alignment, as well as different design choices for alignment, using object localization and detection metrics. Our findings, summarized in Table 3, reveal that image-text alignment plays a critical role in joint optimization, with the choice of alignment method impacting all evaluation metrics. Notably, when joint optimization is applied, incorrect labeling using vanilla CLIP significantly impacts CorLoc and DecRate, while alignment using patch-based CLIP leads to significant improvements across all three metrics: we can see from the first two rows only mAP is different, and the one with patch-based CLIP is better. These results stress the importance of carefully tuning image-text alignment for optimal performance.

7.1. Part-whole problem with joint optimization

To estimate the effectiveness of slot localization in identifying objects, we assess SO, PO, GO, and BG on both I-VID and Y-VIS datasets, and present the results in Table 4. Since slots are a direct output of video slot learning, they are more inclined to focus on parts of objects, accounting for 24.05% and 34.80% of the results (the PO column), rather than the entire object, which only represents 8.87% and 6.04%. We observe that slots rarely contain several objects (GO column), and instead, often capture background

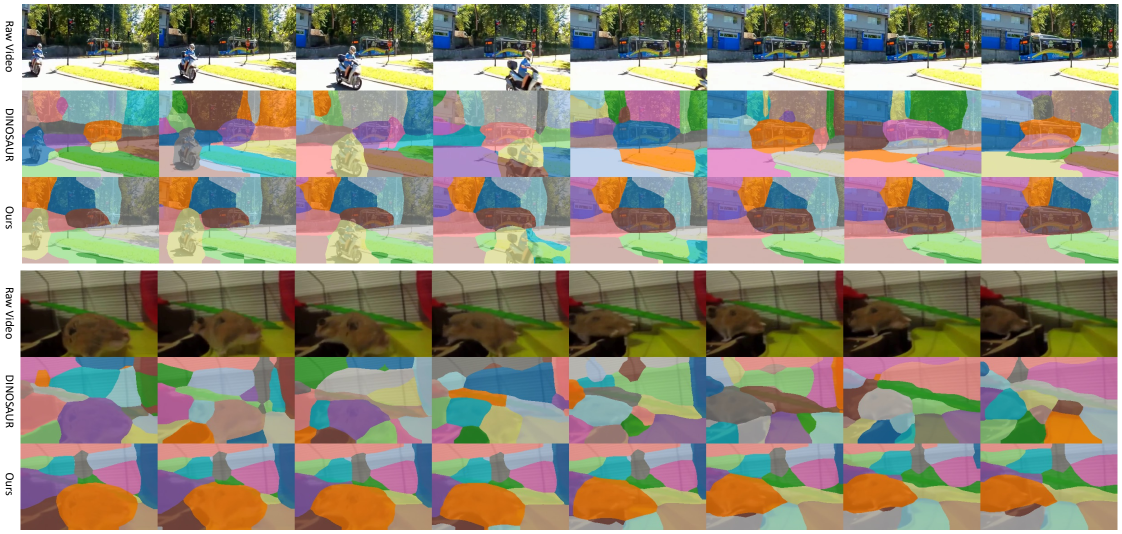


Figure 6. **Consistent localization in video.** We compare the localization performance from DINOSAUR and ours on two 8-frame video samples. We visualize the spatially interpolated localization for patch-based slot assignment. DINOSAUR is an image-based baseline thus its localization has no temporal consistency. In contrast, ours, as a spatio-temporal localization method, provides temporal consistent localization, as the same region or object are consistently localized by the same slot (indicated by the same color) across frames.

Dataset	Joint	CLIP	CorLoc	DecRate	mAP_{50}
I-VID	×	Vanilla	42.99	34.40	0.43
		Patch-based	42.99	34.40	5.20
	✓	Vanilla	15.65	13.73	3.04
		Patch-based	60.90	53.75	29.23
Y-VIS	×	Vanilla	39.36	30.31	0.77
		Patch-based	39.36	30.31	5.96
	✓	Vanilla	25.54	21.27	11.08
		Patch-based	63.74	55.05	35.19

Table 3. **Ablations on patch-based CLIP.** We use the VideoMAE backbone and spatio-temporal video slot to conduct this ablation.

Dataset	Joint	SO	PO	GO	BG
I-VID	×	8.87%	24.05%	1.96%	65.13%
	✓	49.33%	20.42%	7.01%	23.24%
Y-VIS	×	6.04%	34.80%	0.77%	58.39%
	✓	40.75%	28.32%	5.64%	25.29%

Table 4. **Statistics of video slots.** We counted the percentage for each type of slots. **SO**: Single Object. **PO**: Part of an Object. **GO**: Group of Objects. **BG**: Slots for not ***annotated*** regions in the dataset.

classes (65% and 58%; BG column).

The reason for the relatively low proportion of SO can be attributed to the fact that visual similarity and common-fate inductive bias can only offer limited information when describing an object as a whole. Since semantics is not inherently present in the visual modality, it is challenging to capture the full meaning of an object with visual cues alone.

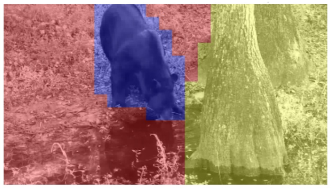
Our joint optimization approach demonstrates a significant increase in the ratio of SO slots. As illustrated in Table 4, the percentage of SO slots rises to 49.33% and 40.75%, respectively, after the optimization with slot labeling. The reduction in PO and BG ratio indicates the source of improvement. First, slots that were previously tracking a significant part of the same object (PO) are merged to form a larger new slot. Further, each BG slot either has no overlap to any object, or only has minor overlap to one object. The reduction in BG ratio indicates that some of the latter case are correctly labeled and eventually merged. This serves as an evidence that the joint optimization directly alleviate the part-whole issue. It should be noted that the proportion of GO also slightly increases, indicating that the model occasionally merge slots too aggressively for multiple instances in the same semantic category. This phenomenon again reflects the challenging associated with part-whole hierarchies.

Regarding the accuracy of labeling, Table 3 presents the mAP score, which takes into account both localization and semantic labeling performance. With the aid of named localization and joint optimization, the model achieves a substantial increase in mAP score of around 24 ($5.20 \rightarrow 29.23$) and 30 ($5.96 \rightarrow 35.19$), respectively. Although these results fall short of the SOTA performance achieved by models using labels [3, 10], considering that our framework is entirely unsupervised, the results are encouraging.

7.2. Qualitative case studies

Spatio-temporal video slots. Our model offers a significant advantage over ordinary image slots in the form of spatio-temporal video slots. These slots consistently focus on the same part across frames, which ensures temporal consistency and improves object-centric localization in videos. Figure 6 provides visualizations that highlight the differences between image-based and video-based object-centric localization on two videos. While DINOSAUR can localize objects at a certain level in each frame, its results do not account for temporal consistency. This is evident as one object may be localized by a different number of slots among frames, and the background area may contain a totally different slot segmentation. In contrast, our model produces consistent localization across frames for all slots, including the background ones. This demonstrates that video slots tend to track all content in a short video clip. Moreover, this tracking behavior also handles objects that move in and out of frames. In the first sample, the biker disappears after frame 5, and its yellow slot never appears again. Similarly, the house in the background gradually appears and occupies the purple slot from frame 3.

Patch-based CLIP. Two qualitative examples comparing the labeling results from patch-based CLIP and vanilla CLIP are shown in Figure 7. The results clearly demonstrate that patch features from vanilla CLIP are inadequate for direct labeling algorithm, whereas the patch-based CLIP provides much better alignment and improved performance.



Vanilla CLIP: red 34% tree, blue 42% tree, yellow 11% bear.
Patch-based CLIP: red 15% water, blue 98% bear, yellow 99% tree.

Figure 7. **Comparing patch-based CLIP and vanilla CLIP**. Vanilla CLIP cannot find the right name for the visual content in each slot, while patch-based CLIP names each slot correctly.

Labeling any slot. Given an appropriate list of semantics, our approach has the potential to name any slot. Figure 8 shows a single-frame example of the localization from video slots and their corresponding names. The result contains not only correct localization and labeling for the central objects, but also with reasonable names for other slots. Inevitably, there are slots that are not easy to name as their semantic features are not similar to any text features. Reasons of low similarity include 1) the precise name of the visual content is not included in the semantic list, 2) the visual content is a composition of multiple semantics (e.g. the red slot is a combination of leaves, road and fence), and 3) the visual content is incomplete or ambiguous (e.g. the light green slot only shows part of a building). Nevertheless, this example demonstrates both the potential of our model for open-set detection and segmentation, and at the same time also the challenging nature of these downstream tasks under an unsupervised setting.

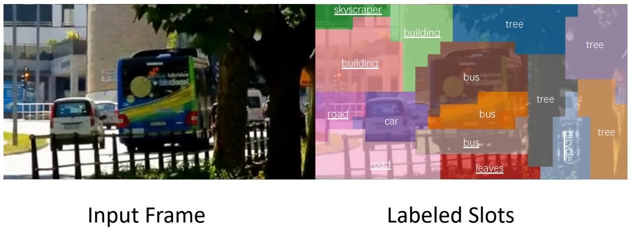


Figure 8. **Labeling any slot**. The first image shows the input frame. The second image shows the localization of that frame, as well as names for all slots. Slots names with underscore are those having a low cosine similarity.

8. Discussion

On Supervision Signal. Our video object localization is entirely unsupervised apart from the implicit annotation contained in CLIP. Besides self-supervised pretraining and video slot learning, the patch-based CLIP finetuning is also without human annotation. Further, it is worth noting that either the pretrained or finetuned CLIP model do not generate any pseudo annotation to guide the learning process of the framework. The patch-based CLIP model is only used for label assignment at inference stage.

Future work. Handling long videos is undeniably an intriguing task, but its comprehensive treatment lies beyond the scope of current work. However, we foresee potential extensions of our method from an engineering perspective in future research. For instance, the model can be readily expanded from processing a fixed number of frames to accommodating a larger and more flexible range of frames with relatively minor adjustments, such as position embedding interpolation or temporal frame sub-sampling.

9. Conclusion

In this paper, we proposed a novel unsupervised method for object localization in videos that leverages the power of slot attention combined with a pre-trained CLIP image-text model. By fine-tuning the last layer of CLIP in a self-supervised manner on unlabeled image data, we established a means for applying the image-text mapping to local patch tokens. Through a joint optimization process with both visual and semantic information, we achieved, for the first time, high-quality and spatio-temporally consistent object localization outcomes on real-world video datasets. While the current coarse localization provides an encouraging starting point, we acknowledge that it still suffers from low resolution of the patch tokens. Moreover, to complete our object detection and tracking pipeline, we are yet to move beyond semantic localization and accurately differentiate between individual object instances. Nevertheless, we are optimistic that our findings will pave the way for label-free training on complex video analysis tasks

References

[1] Zhipeng Bao, Pavel Tokmakov, Allan Jabri, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Discovering objects that can move. In *CVPR*, 2022.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. *ICCV*, 2021.

[3] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory Enhanced Global-Local Aggregation for Video Object Detection. In *CVPR*, 2020.

[4] Andrea Dittadi, Samuele Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and Robustness Implications in Object-Centric Learning. In *ICML*, 2022.

[5] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, 2022.

[6] Gamaleldin Fathy Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael Curtis Mozer, and Thomas Kipf. SAVi++: Towards End-to-End Object-Centric Learning from Real-World Videos. In *NeurIPS*, 2022.

[7] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations. In *ICLR*, 2020.

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

[9] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. In *ICCV*, 2017.

[10] Lu He, Qianyu Zhou, Xiangtai Li, Li Niu, Guangliang Cheng, Xiao Li, Wenxuan Liu, Yunhai Tong, Lizhuang Ma, and Liqing Zhang. End-to-End Video Object Detection with Spatial-Temporal Transformers. In *International Conference on Multimedia*, 2021.

[11] Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional Object-centric Learning from Video. In *ICLR*, 2022.

[12] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. *arXiv:2210.04150*, 2022.

[13] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention. In *NeurIPS*, 2020.

[14] 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.

[15] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. *arXiv:2212.04994*, 2022.

[16] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748*, 2018.

[17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. In *ICML*, 2021.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[19] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *ICLR*, 2023.

[20] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[21] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple Unsupervised Object-Centric Learning for Complex and Naturalistic Videos. In *NeurIPS*, 2022.

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

[23] Huy V. Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *ECCV*, 2020.

[24] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew-Soon Ong, and Chen Loy Change. Unsupervised Object-Level Representation Learning from Scene Images. In *NeurIPS*, 2021.

[25] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. GroupViT: Semantic Segmentation Emerges from Text Supervision. In *CVPR*, 2022.

[26] Andrii Zadaianchuk, Matthaeus Kleindessner, Yi Zhu, Francesco Locatello, and Thomas Brox. Unsupervised Semantic Segmentation with Self-supervised Object-centric Representations. In *ICLR*, 2023.