

# In-N-Out Generative Learning for Dense Unsupervised Video Segmentation

Xiao Pan<sup>1,2</sup>, Peike Li<sup>2,3</sup>, Zongxin Yang<sup>1</sup>, Huiling Zhou<sup>2</sup>, Chang Zhou<sup>2</sup>,  
Hongxia Yang<sup>2</sup>, Jingren Zhou<sup>2</sup>, Yi Yang<sup>1</sup>

<sup>1</sup>ReLER Lab, CCAI, Zhejiang University, China

<sup>2</sup>Alibaba DAMO Academy, China

<sup>3</sup>Australian Artificial Intelligence Institute, University of Technology Sydney, Australia

## ABSTRACT

In this paper, we focus on unsupervised learning for Video Object Segmentation (VOS) which learns visual correspondence (*i.e.*, the similarity between pixel-level features) from unlabeled videos. Previous methods are mainly based on the contrastive learning paradigm, which optimize either in image level or pixel level. Image-level optimization (*e.g.*, the spatially pooled feature of ResNet) learns robust high-level semantics but is sub-optimal since the pixel-level features are optimized implicitly. By contrast, pixel-level optimization is more explicit, however, it is sensitive to the visual quality of training data and is not robust to object deformation. To complementarily perform these two levels of optimization in a unified framework, we propose the In-aNd-Out (INO) generative learning from a purely generative perspective with the help of naturally designed class tokens and patch tokens in Vision Transformer (ViT). Specifically, for image-level optimization, we force the out-view imagination from local to global views on class tokens, which helps capture high-level semantics, and we name it as out-generative learning. As to pixel-level optimization, we perform in-view masked image modeling on patch tokens, which recovers the corrupted parts of an image via inferring its fine-grained structure, and we term it as in-generative learning. To discover the temporal information better, we additionally force the inter-frame consistency from both feature and affinity matrix levels. Extensive experiments on DAVIS-2017 val and YouTube-VOS 2018 val show that our INO outperforms previous state-of-the-art methods by significant margins. Code: [https://github.com/pansanity666/INO\\_VOS](https://github.com/pansanity666/INO_VOS)

## CCS CONCEPTS

• **Computing methodologies** → **Video segmentation; Image segmentation; Image representations.**

## KEYWORDS

unsupervised video object segmentation, self-supervised learning, dense prediction, generative learning

## ACM Reference Format:

Xiao Pan<sup>1,2</sup>, Peike Li<sup>2,3</sup>, Zongxin Yang<sup>1</sup>, Huiling Zhou<sup>2</sup>, Chang Zhou<sup>2</sup>, Hongxia Yang<sup>2</sup>, Jingren Zhou<sup>2</sup>, Yi Yang<sup>1</sup>. 2022. In-N-Out Generative Learning for Dense Unsupervised Video Segmentation. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22)*, October 10–14, 2022, Lisboa, Portugal. ACM, New York, NY, USA, 13 pages. <https://doi.org/https://doi.org/10.1145/3503161.3547909>

## 1 INTRODUCTION

Video Object Segmentation [1, 22, 24, 32, 38, 40] (VOS) is a fundamental video understanding task having a wide range of real-world applications, *e.g.*, augmented reality [35], and video editing [18]. In this task, we aim to segment a specified object instance throughout the entire video sequence, given only the ground-truth object mask on the first frame. Although the development of convolutional neural networks (CNNs) [11] has significantly advanced the VOS task, the success of these approaches highly relies on the cost-intensive and time-consuming dense mask annotations to train the networks. Moreover, the fully-supervised VOS is also largely limited by the diversity and scale of the annotated datasets. To relieve such limitations, recently unsupervised/self-supervised methods for VOS [1, 12, 15] have drawn considerable attention, which can completely liberate the request for mask annotations.

However, the previous approaches still suffer from the following perspectives: (i) *Image-level vs. pixel-level*. Several existing methods [12, 36] perform self-supervised learning on the image-level features, *e.g.*, the spatially pooled feature of ResNet [11]. Such optimization can learn robust high-level semantics, however, it is sub-optimal since the pixel-level features used for calculating the final correspondence are learned implicitly. Another line of works [1, 15, 33] explicitly optimize on the pixel-level features, however, such paradigm may be sensitive to the visual quality of training data and is lack of the high-level semantic scope which may reduce the robustness toward deformation. We believe that these two levels of optimization are complementary to each other and can be integrated into a unified framework; (ii) *CNN vs. ViT*. Previous unsupervised methods [1, 12, 36] for VOS adopt the convolutional networks (CNNs) [11] as the backbone model. However, we propose that Vision Transformer (ViT) [6, 28] is a better choice for unifying these two levels of optimization. Specifically, different from the CNNs which obtain the image-level features via the average pooling of pixel-level features, the naturally designed class tokens and patch tokens of ViT have their own semantic meanings, *i.e.*, class tokens capture the high-level semantics which are suitable for image-level optimization, while patch tokens represent fine-grained details, which are appropriate for pixel-level optimization.

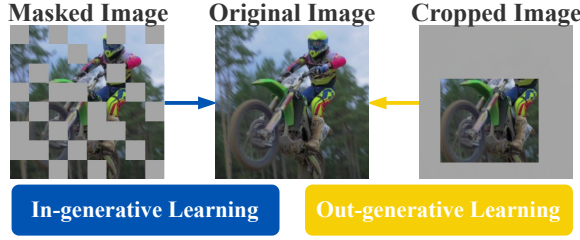
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

MM '22, October 10–14, 2022, Lisboa, Portugal.

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9203-7/22/10...\$15.00

<https://doi.org/https://doi.org/10.1145/3503161.3547909>



**Figure 1: Idea illustration for In-N-Out generative learning. In-generative learning is the in-view recovery of masked patches, while out-generative learning is the out-view imagination from local to global views.**

(iii) *Contrastive vs. generative.* Previous self-supervised methods for VOS [1, 12, 36] employ a self-training objective mainly based on contrastive formulation [10]. However, their experimental results show unsatisfactory scalability toward the training data scales, especially for the pixel-level methods [1]. On the other hand, the recent success of the generative learning paradigm [9] shows its better scalability and robustness. Therefore, we propose that solving the VOS task from the generative learning perspective is promising, yet unexplored.

To tackle the aforementioned issues, we present a simple yet effective framework called In-aNd-Out (INO) generative learning from a novel fully generative learning perspective. As opposed to the previous methods [1, 12, 36] (detailed in Table 1), INO integrates image-level and pixel-level optimization in a unified framework with the help of naturally designed high-level class tokens and fine-grained patch tokens in ViT.

More concretely, as illustrated in Fig. 1, our INO framework contains two generative objectives, *i.e.*, in-generative learning and out-generative learning. (i) *Out-generative learning* is the *out-view* imagination from local views to global views on class tokens, which corresponds to the image-level optimization. We aim to learn high-level visual semantics via imagining the global visual information given only a portion of the local visual part, which may improve the robustness toward deformation and occlusion. (ii) *In-generative learning* is the *in-view* recovery of randomly masked patch tokens in the feature embedding space, which belongs to the pixel-level optimization. The goal of in-generative learning is to capture fine-grained structural information, which is beneficial for recognizing the gradually appearing parts of a semantic object given incomplete labels in the first frame. In this way, both in-and-out generative objectives complement each other towards better visual representation learning. To better discover the temporal information, we additionally equip INO with temporally-persistent constraints, by forcing the inter-frame consistency from both feature level and affinity matrix level. Extensive experiments on DAVIS-2017 val and YouTube-VOS 2018 val show that our INO outperforms previous methods by significant margins. Our main contributions are summarized as follows,

- We propose a simple yet effective framework INO to tackle the challenging unsupervised learning for VOS, which integrates image-level and pixel-level optimization in a unified framework by leveraging the structural superiority of ViT.
- To the best of our knowledge, we make the first attempt to conduct unsupervised learning for VOS from a novel fully

**Table 1: Comparisons between different state-of-the-art VOS methods.**

Method	Self-supervised	Transformer-based	Unified optimization	Generative learning
AOT (NIPS 2021) [41]	✗	✓	✗	✗
CRW (NIPS 2020) [12]	✓	✗	✗	✗
VFS (ICCV 2021) [36]	✓	✗	✗	✗
DUL (NIPS 2021) [1]	✓	✗	✗	✗
INO (Ours)	✓	✓	✓	✓

generative learning perspective based on the idea of masked image modeling.

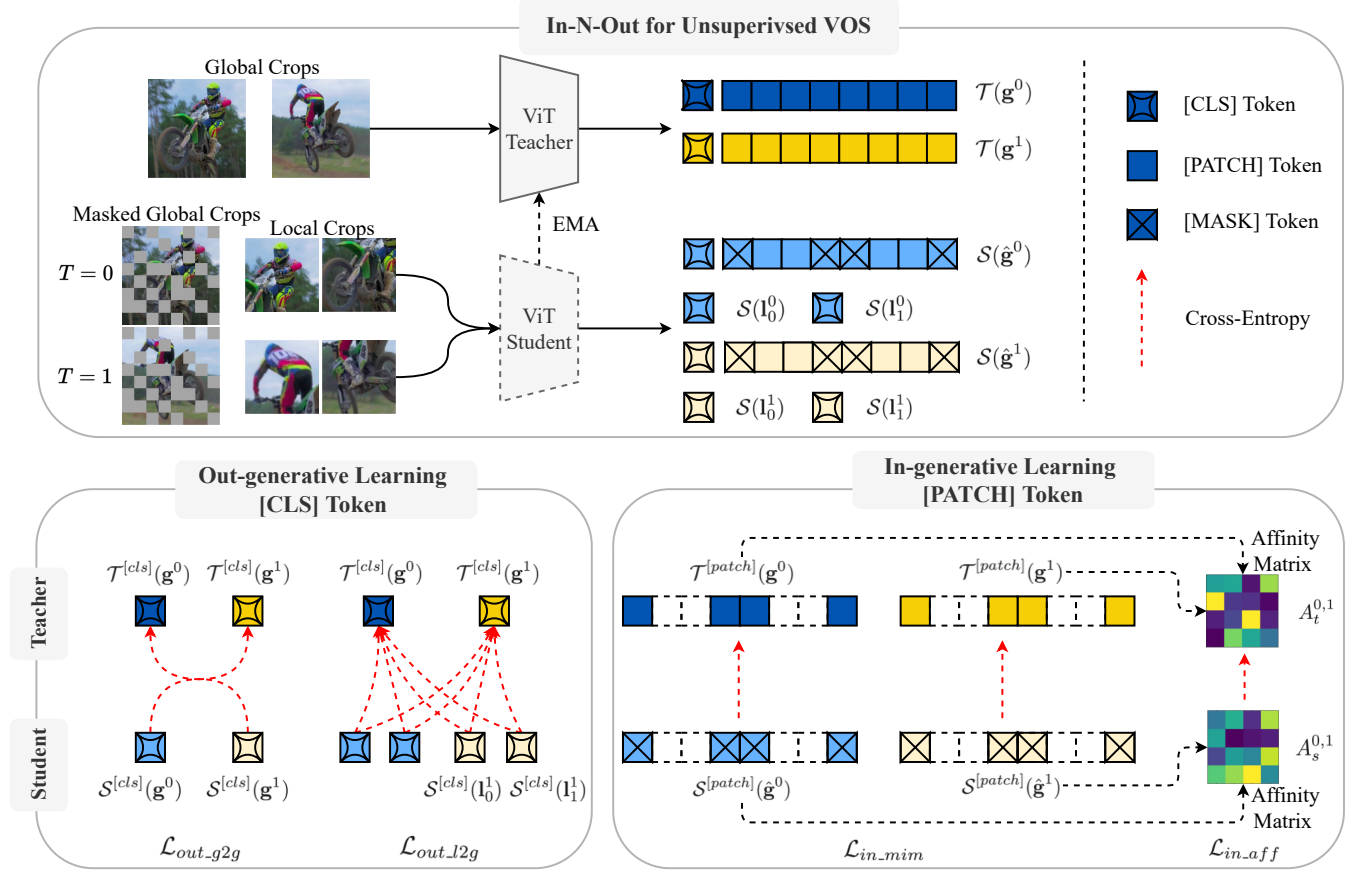
- We attain a new state-of-the-art performance on unsupervised learning for VOS.

## 2 RELATED WORK

**Vision Transformers for Dense Prediction.** Most previous VOS methods [1, 12, 15, 33, 36] adopt convolutional neural networks (CNNs) as the architectures to learn the visual representations. Recently, the development [6, 28] of Vision Transformer (ViT) architecture shows overwhelming success over CNNs. Beyond the simple image classification tasks [19], researchers have successfully adopted the ViT architecture in several dense prediction tasks, *e.g.*, object detection [3], semantic segmentation [34], *etc.* However, these works mainly focus on still images and study the fully-supervised settings. In contrast, our INO is specially designed to solve the video object segmentation via self-supervised training. We exploit the architecture privilege of ViT by using the naturally designed high-level class tokens and fine-grained patch tokens.

**VOS via Self-supervised Learning.** Some previous approaches [17, 22, 24, 39, 42] learn visual representation fully supervised by annotated datasets with pixel-wise labels. However, these fully-supervised VOS methods are highly restricted by the diversity of annotated categories and the scale of the annotated dataset. In this work, we explore a more promising and efficient way which is to leverage self-supervised learning [4, 10] to conduct VOS. Previous self-supervised works either only consider the high-level global representations [12, 36], or directly perform the contrastive learning on the fine-grained pixel-wise features [1]. Benefiting from the flexibility of vision transformer-based architecture, we optimize both high-level semantics on class tokens and fine-grained semantics on patch tokens, which brings better correspondence learning.

**From Contrastive Learning to Generative Learning.** Recently, contrastive learning [8, 10] has been popular for self-supervised learning, which models image similarity and dissimilarity between different views. Although migrating these methods on VOS tasks [1, 12, 13, 36, 43] has achieved a preliminary success, the contrastive-based methods strongly depend on data augmentation [12], and may also suffer the scalability problem [1, 12]. Inspired by masked language modeling [5] in NLP, recent attempts [2, 9] learn representations by masked image modeling. However, these methods either reconstruct the original images in original pixels [9], or predict the quantized discrete tokens [2]. As illustrated in Fig. 1, we discover more abundant supervised signals directly in feature embedding space, by recovering masked patch tokens (in-generative) and imagining global information (out-generative). To the best of our knowledge, we make the first attempt to address self-supervised



**Figure 2: Overview of INO framework.** For illustration, we assume that the video length  $L = 2$ , and for each frame we crop 1 global crop and 2 local crops. (i) *Out-generative learning* is the *out-view* mapping between high-level class tokens, which is composed of  $\mathcal{L}_{out\_g2g}$  and  $\mathcal{L}_{out\_l2g}$ . Concisely,  $\mathcal{L}_{out\_g2g}$  is cross-frame calculated between global crop outputs, while  $\mathcal{L}_{out\_l2g}$  is all the possible local-to-global mappings. (ii) *In-generative learning* is the *in-view* recovery on fine-grained patch tokens of global crops, which is composed of  $\mathcal{L}_{in\_mim}$  and  $\mathcal{L}_{in\_aff}$ . In detail,  $\mathcal{L}_{in\_mim}$  is the mapping between mask tokens and the corresponding teacher patch tokens, while  $\mathcal{L}_{in\_aff}$  improves the inter-frame correspondence via bootstrapping.

learning for VOS from a fully generative learning perspective based on the idea of masked image modeling.

### 3 METHOD

**Overview.** Unsupervised learning for video object segmentation targets on achieving semantically discriminative representations via training on unlabeled videos. During the inference stage, given the mask annotation of the first frame, the label propagation is performed via the correspondence (similarity) between the extracted feature maps [1, 12, 15, 36]. As illustrated in Fig. 2, we introduce the proposed In-N-Out generative learning framework for the unsupervised learning for VOS, which is composed of out-generative learning on high-level class tokens and in-generative learning on fine-grained patch tokens. In § 3.1 we first introduce the generic teacher-student framework. § 3.2 demonstrates the proposed out-generative learning targets on mining high-level semantic consistency via the imagination between augmented crops. § 3.3 introduces the proposed in-generative learning which focuses on achieving fine-grained semantics via the recovery of the corrupted

semantic structure. We present the brief training and inference pipeline of the proposed INO in § 3.4.

#### 3.1 Generative Learning Framework

We use the commonly used teacher-student framework [4, 44] for self-supervised learning. The teacher  $\mathcal{T}$  and student  $\mathcal{S}$  share the same architecture which includes a backbone (e.g., ViT [6, 28]) and a projection head. Without loss of generality, we directly adopt the original ViT implementation. The improvement of the backbone model is not the focus of this paper. The teacher parameters are the Exponentially Moving Average (EMA) of the student parameters. To avoid collapse, a stop-gradient operator is applied to the teacher, and the teacher output is centered by the computed batch mean [4]. Then, each network outputs (including both class tokens and patch tokens) are normalized with a temperature softmax to get the final categorical distributions [44]. The output categorical distributions of the teacher are taken as the generation target of the student output and their similarity is measured with a standard cross-entropy loss. Notably, the class tokens mainly capture the

high-level semantic information, while the patch tokens focus more on the fine-grained details. Therefore, we choose class tokens for out-generative learning and patch tokens for in-generative learning, respectively.

### 3.2 Out-generative Learning

Given the  $i$ -th frame in a video clip of length  $L$ , we first achieve one global crop  $\mathbf{g}^i$  and  $M$  local crops  $\{\mathbf{l}^{ij}\}_{j=1}^M$  via *random resized cropping* under different scales. Then, *flipping* and *color jitter* are randomly applied for each crop. The augmented crops from frames of the same video clip may be quite different in low-level vision due to the performed augmentation and the motion nature of videos. However, since they are from the same semantic scene, they share similar high-level semantic meaning. We intend to leverage such high-level semantic consistency as the supervision signal for out-generative learning, which can be further divided into global-to-global and local-to-global generation, as illustrated in the lower-left part of Fig. 2.

**Temporal Consistency via Global-to-global Generation.** Considering the consecutive nature of video sequences, the global crops from different frames of a certain sequence may share most of the semantic objects when the scale range for cropping is large. In this case, the most salient difference may come from the motion of objects, which should be recognized for better segmentation label propagation, *e.g.*, the person with different poses in different frames should be recognized as the same person. Therefore, we propose to perform generation between these inter-frame global crops to capture such temporal semantic consistency in the presence of motion.

Specifically, for a sequence of length  $L$ , we empirically split the sequence into halves and then zip them as frame pairs  $\mathcal{P} = \{(n, L/2 + n)\}_{n=1}^{L/2}$ . Taking  $L = 6$  as an example, the formulated pairs are  $\{(1, 4), (2, 5), (3, 6)\}$ . Given a frame pair  $(i_1, i_2) \in \mathcal{P}$ , all the global crops  $\{\mathbf{g}^{i_1}, \mathbf{g}^{i_2}\}$  are sent to the student together with the teacher, and the global-to-global out-generative learning is performed as:

$$\mathcal{L}_{out\_g2g} = \frac{1}{|\mathcal{P}|} \sum_{(i_1, i_2) \in \mathcal{P}} \sum_{\mathbf{t} \in \mathbb{T}} \sum_{\mathbf{s} \in \mathbb{S} \setminus \{\mathbf{t}\}} -\mathcal{T}^{[cls]}(\mathbf{t})^T \log \mathbf{S}^{[cls]}(\mathbf{s}), \quad (1)$$

where  $\mathbb{T} = \{\mathbf{g}^{i_1}, \mathbf{g}^{i_2}\}$ ,  $\mathbb{S} = \{\mathbf{g}^{i_1}, \mathbf{g}^{i_2}\}$ .

**Semantic Correlation via Local-to-global Generation.** Compared with the global-to-global generation, local-to-global generation solves a harder task, which is to imagine the complete scene given limited information from random semantic fragments. In this scenario, not only motion consistency, but also the inference of the high-level semantic correlation is required. For example, given the fragment of a motorcross and a part of the human leg, the model is required to infer that “a rider is riding a motorcross”, as illustrated in Fig. 1. Such high-level inference helps the model to capture more robust and complete semantics, which is beneficial to the stable propagation of masks.

Specifically, for frame pair  $(i_1, i_2) \in \mathcal{P}$ , all the local crops are sent to the student and calculated with the previous teacher outputs of global crops. Similar to Eq 1, the local-to-global generative learning

is performed as:

$$\mathcal{L}_{out\_l2g} = \frac{1}{|\mathcal{P}|} \sum_{(i_1, i_2) \in \mathcal{P}} \sum_{\mathbf{t} \in \mathbb{T}} \sum_{\mathbf{s} \in \mathbb{S}} -\mathcal{T}^{[cls]}(\mathbf{t})^T \log \mathbf{S}^{[cls]}(\mathbf{s}), \quad (2)$$

where  $\mathbb{T} = \{\mathbf{g}^{i_1}, \mathbf{g}^{i_2}\}$ ,  $\mathbb{S} = \{\mathbf{l}^{i_1j}, \mathbf{l}^{i_2j}\}_{j=1}^M$ .

### 3.3 In-generative Learning

The out-generative learning leverages the class tokens to capture the high-level semantic consistency, however, the dense segmentation task requires more fine-grained semantic information to precisely capture the correspondence. Therefore, we introduce the in-generative learning on patch tokens of global crops, which is composed of intra-frame masked image modeling and inter-frame affinity consistency, as shown in the lower-right part of Fig. 2.

**Intra-frame Masked Image Modeling.** Inspired by [2, 44], we first perform blockwise masking on the input global crops of the *student* module (as illustrated in Fig. 2). Specifically, assuming that a global crop from the  $i$ -th frame  $\mathbf{g}^i$  is split into  $P$  tokens  $\mathbf{g}^i = \{\mathbf{g}_j^i\}_{j=1}^P$ , we then randomly choose a ratio  $r$  as the proportion of masked tokens and get a random mask  $\mathbf{m} \in \{0, 1\}^P$ . According to  $\mathbf{m}$ , the picked  $K = P \cdot r$  tokens  $\{g_i | m_i = 1\}$  are replaced with a global learnable mask token and get a corrupted token sequence  $\hat{\mathbf{g}}^i$ . After that,  $\hat{\mathbf{g}}^i$  is sent to the following attention module and prediction head for categorical distributions output. For the *teacher* module, the original unmasked token sequence  $\mathbf{g}^i$  is kept as input. Finally, similar as Eq. 2, we take the corresponding teacher outputs of the masked tokens as the generation target and calculate the cross-entropy between them for each global crop in a video clip:

$$\mathcal{L}_{in\_mim} = \frac{1}{L} \sum_{i=1}^L \sum_{j=1}^P m_j \cdot -\mathcal{T}_j^{[patch]}(\mathbf{g}^i)^T \log \mathbf{S}_j^{[patch]}(\hat{\mathbf{g}}^i). \quad (3)$$

Intuitively, with Eq. 3, we force the model to generate the corrupted patches based on the limited semantic information from the reserved patches. Such supervision forces the model to achieve the complete and fine-grained semantics, which is important for recognizing the gradually appearing parts given incomplete labels in the first frame. For example, the figure of the person may be incomplete (*i.e.*, corrupted) at the first time of appearance due to the occlusion or limited shooting angle. The model should possess the ability to recognize the rest part of a person which may gradually appear in the following frames, and this is in line with the target of  $\mathcal{L}_{in\_mim}$ .

**Inter-frame Affinity Consistency.**  $\mathcal{L}_{in\_mim}$  explores the fine-grained semantic information spatially for each global crop, however, the temporal fine-grained semantic consistency between frames should also be considered, therefore, we propose the affinity consistency constrain as follows.

Given the  $i$ -th global crop in a video sequence of length  $L$ , we represent the  $l_2$ -normalized  $d$ -dimensional distribution matrix of the masked tokens from the teacher module as  $Q_t^i \in \mathbb{R}^{K \times d}$ . Similarly, the corresponding distribution matrix for the student module is represented as  $Q_s^i \in \mathbb{R}^{K \times d}$ . Then, the affinity matrix from timestep  $i$  to  $i + 1$  for the teacher outputs is calculated as:

$$A_t^{i,i+1} = \text{softmax}((Q_t^i Q_t^{i+1}^T) / \tau_t), \quad (4)$$



**Table 2: Comparisons with state-of-the-art methods on DAVIS-2017 val. RN-18 and ViT-S/8 represent for ResNet-18 and ViT-Small with a patch size of 8, separately. “†” means using 2 times larger resolution for inference. We also report the number of video sequences (N) and the total video duration (T) for each dataset.**

Method	Arch	Dataset	N/T	$\mathcal{J} \& \mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{F}_m$	$\mathcal{F}_r$
TimeCycle [33]	RN-18	VLOG	114K / 344h	-	40.1	-	38.3	-
TimeCycle [33]	RN-50	VLOG	114K / 344h	-	41.9	-	39.4	-
CorrFlow†[16]	RN-18	Kinetics	300K / 833h	49.5	47.7	53.2	51.3	56.5
CorrFlow†[16]	RN-18	OxUvA	366 / 14h	50.3	48.4	53.2	52.2	56.0
ContCorr [31]	RN-18	TrackingNet	30K / 140h	63.0	60.5	-	65.5	-
MAST†[15]	RN-18	OxUvA	366 / 14h	63.7	61.2	73.2	66.3	78.3
MAST†[15]	RN-18	YT-VOS	4.5K / 5h	65.5	63.3	73.2	67.6	77.7
CRW [12]	RN-18	Kinetics	300K / 833h	67.6	64.8	76.1	70.2	82.1
JSTG [43]	RN-18	Kinetics	300K / 833h	68.7	65.8	77.7	71.6	84.3
DUL [1]	RN-18	OxUvA	366 / 14h	65.3	63.4	76.1	67.2	79.7
DUL [1]	RN-18	Kinetics	300K / 833h	68.7	66.7	81.4	70.7	84.1
DUL [1]	RN-18	YT-VOS	4.5K / 5h	69.3	67.1	81.2	71.6	84.9
DUL [1]	RN-18	TrackingNet	30K / 140h	69.4	67.1	80.9	71.7	84.8
VFS [36]	RN-18	Kinetics	300K / 833h	67.6	64.8	-	70.2	-
VFS [36]	RN-50	Kinetics	300K / 833h	69.4	66.7	-	72.0	-
<b>INO (Ours)</b>	ViT-S/8	Charades	10K / 82h	67.0	63.7	72.7	70.4	82.9
<b>INO (Ours)</b>	ViT-S/8	Kinetics	300K / 833h	<b>72.5</b>	<b>68.7</b>	<b>82.0</b>	<b>76.3</b>	<b>89.0</b>

**Table 3: Comparisons with state-of-the-art methods on YouTube-VOS 2018 val benchmark. The object classes in the val set are partly overlapped with the *training* set, therefore the performance is distinguished as “seen” and “unseen” categories. All results are evaluated and reported through the online testing server [37].**

Method	Dataset	Mean	Seen		Unseen	
			$\mathcal{J}_m$	$\mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
Colorize [30]	Kinetics	38.9	43.1	38.6	36.6	37.4
CorrFlow†[16]	OxUvA	46.6	50.6	46.6	43.8	45.6
MAST†[15]	YT-VOS	64.2	63.2	64.9	60.3	67.7
CRW [12]	Kinetics	69.9	68.7	70.2	65.4	75.2
DUL [1]	YT-VOS	69.9	69.6	71.3	65.0	73.5
DUL [1]	Kinetics	70.6	69.9	71.3	<b>66.5</b>	74.8
DUL [1]	TrackingNet	70.7	70.2	71.9	66.3	74.5
<b>INO (Ours)</b>	Kinetics	<b>71.3</b>	<b>70.7</b>	<b>73.2</b>	65.6	<b>75.6</b>

where  $A_t^{i,i+1} \in \mathbb{R}^{K \times K}$  and  $\tau_t$  is the temperature. Similarly, the corresponding affinity matrix and temperature for the student outputs are  $A_s^{i,i+1}$  and  $\tau_s$ , respectively. Then, we calculate the cross-entropy between these two affinity matrices as follows:

$$\mathcal{L}_{in\_aff} = \frac{1}{L-1} \sum_{i=1}^{L-1} \sum_{j=1}^K -A_t^{i,i+1}[j,:]^T \log A_s^{i,i+1}[j:], \quad (5)$$

where  $A_s^{i,i+1}[j,:]$  represents for the  $j$ -th row vector of the matrix, which is the softmax normalized cosine similarity between the  $j$ -th mask token of frame  $i$  and all the  $K$  masked tokens of frame  $i+1$ , i.e., the correspondence. Intuitively, we intend to learn the fine-grained temporal correspondence via bootstrapping, which is beneficial for the ultimate goal of propagating segmentation labels.

### 3.4 Training & Inference Pipeline

In this subsection, we introduce the whole pipeline for INO to achieve unsupervised learning for VOS.

During the training stage, we start by training the ViT backbone with the raw data from video recognition datasets Kinetics-400 [14] and Charades [27] without using any human annotation labels. Our network is trained in a self-supervised manner with learning objectives:

$$\mathcal{L}_{INO} = \mathcal{L}_{out\_g2g} + \mathcal{L}_{out\_l2g} + \mathcal{L}_{in\_mim} + \mathcal{L}_{in\_aff}. \quad (6)$$

To maintain simplicity, here we treat all these terms with equal contributions. Once the backbone model is trained, we can evaluate directly on the DAVIS-2017 val [26] and YouTube-VOS 2018 val [37] without fine-tuning.

During inference, the segmentation label of the first frame is provided and then propagated toward the following frames based on the similarity between extracted feature maps. For fair comparison, we use the same label propagation strategy as [1, 4, 12, 36], detailed in the supplementary material.

## 4 EXPERIMENT

### 4.1 Experimental Settings

**Datasets.** In order to validate the scalability of the proposed INO, we conduct experiments on two large-scale datasets, including Charades [27] and Kinetics-400 [14]. Charades [27] dataset spans 9848 videos with an average length of 30s and records the causal everyday activities at home. Kinetics [14] dataset contains significantly more video sequences (around 230K videos with 10s per video on average). Note that we directly use raw data from the video dataset, no human annotations are involved during the training process. **Evaluation Metrics.** To verify the generalization ability of INO, we benchmark on two challenging video object segmentation benchmarks: DAVIS-2017 val [26] and YouTube-VOS 2018 val [37]. DAVIS-2017 val contains 30 videos in 480p, and YouTube-VOS 2018 val spans 474 videos, and over 90% of them are in 720p. Following previous works [1, 12, 36], we use region similarity ( $\mathcal{J}$ ) and contour accuracy ( $\mathcal{F}$ ) as the evaluation metric [25].

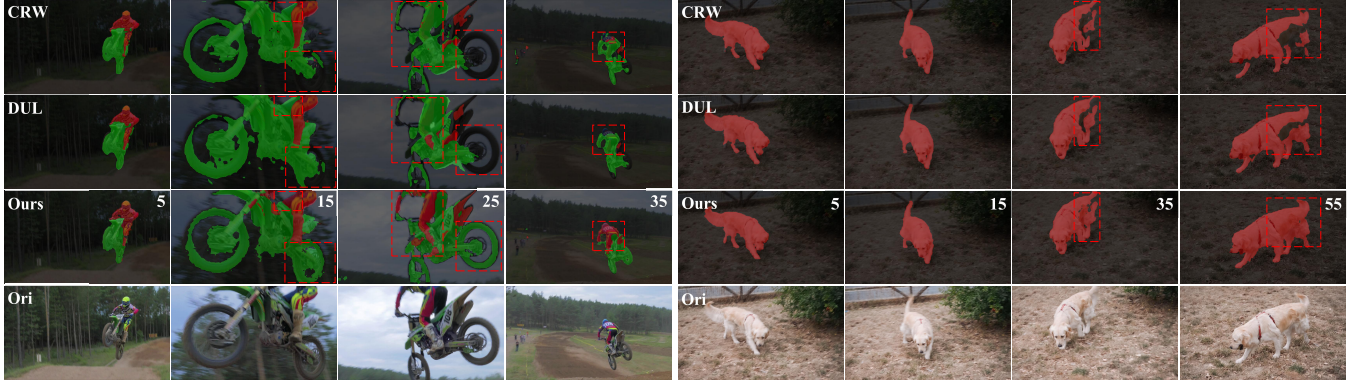


Figure 3: Qualitative comparisons on DAVIS-2017 val. We also display the examples of DUL [1] and CRW [12]. The frame number is illustrated in the upper-right corner. We mark the salient parts where our method performs better with red dotted boxes.

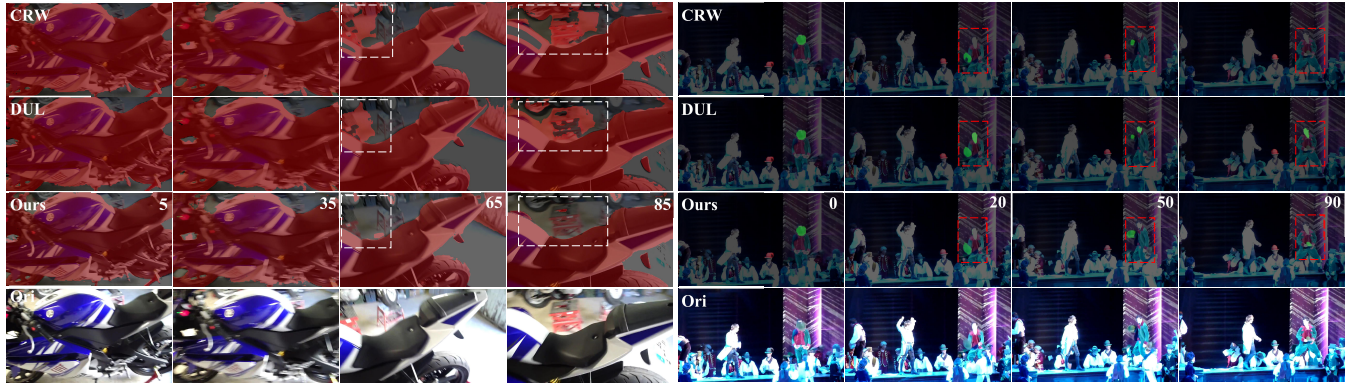


Figure 4: Qualitative comparisons on YouTube-VOS 2018 val. Notably, in the second “hat-trick” case, we illustrate a hard scenario where the hat is rapidly moving with highly blurry. CRW [12] lost the hat gradually, while DUL [1] mistracks toward the human face. Only our INO succeeds to track the hat in motion, which shows that our method can precisely capture the fine-grained correspondence.

**Implementation Details.** Our INO framework adopts ViT as the backbone model. For fair comparison with the competing methods [1, 12, 36] that downsample the input resolution for 8 times during inference, we use the ViT-S/8 configuration by default, which possesses comparable parameters as ResNet-50 [11]. For each iteration, we do in-generative learning with a random probability of 0.5, and  $r$  is randomly sampled from a uniform distribution from 0.1 to 0.5. The temperature is set as  $\tau_s = 0.1$  and  $\tau_t = 0.04$ , separately. The scale range for global and local crops are (0.05, 0.8) and (0.8, 0.95), respectively. The global crops are resized to  $224 \times 224$  while the local crops are resized to  $64 \times 64$ . The number of local crops  $M$  is set as 8, and the video length  $L$  is set as 4. We train INO for 25 epochs on both Charades and Kinetics-400 with 4 and 8 V100 (16GB) GPUs, respectively. Previous works [1, 12, 36] use the last-block output of ResNet for training, while the middle-block output (e.g., *res 3* block in [12]) for inference. Similarly, we use the output of the last layer (i.e., the 12-th layer) of ViT for training and empirically use

the output of the 7-th layer for inference. More implementation details can be found in the supplementary material.

## 4.2 Quantitative Comparisons with State-of-the-art

**Results on DAVIS-2017 benchmark.** In Table 2, we report the performance comparisons between our INO and other competing state-of-the-art methods on DAVIS-2017 val benchmark. In a fair comparison where all the methods use the large-scale Kinetics dataset for training, our INO significantly outperforms other state-of-the-art methods by a large margin. For instance, we outperform the second best VFS [36] by 3.1% in terms of  $\mathcal{J} \& \mathcal{F}_m$  (72.5% vs. 69.4%). Notably, MAST [15] and CorrFlow [16] adopt 2 times larger image resolution than ours during inference, which leads to larger memory footprint. However, our INO still outperforms MAST [15] by 7% on  $\mathcal{J} \& \mathcal{F}_m$  (72.5% vs. 65.5%).

**Better Scalability.** Scalability is an important criterion for self-supervised learning methods. However, as shown in Table 2, the

performance of CorrFlow, MAST, and DUL is not positively correlated with the scale of the training dataset. And the best performance is not achieved on the largest dataset. For instance, even though the dataset size is two orders of magnitude smaller (300k vs. 4.5k), the performance of DUL trained with YouTube-VOS is still higher than Kinetics by 0.6% in  $\mathcal{J}\&\mathcal{F}_m$  score (68.7 % vs. 69.3 %). Actually, for such methods, the best performance are achieved on the datasets with higher visual quality and cleaner background, e.g., YouTube-VOS [37], TrackingNet [23], and OxUvA [29]. This demonstrates that the previous methods are sensitive to the dataset quality and lack of scalability. In contrast, our INO exhibits strong scalability and robustness owing to the generative learning paradigm, e.g., the performance boosts by 5.5% from Charades to Kinetics.

**Results on YouTube-VOS 2018 benchmark.** Following [1, 15], we additionally evaluate INO on YouTube-VOS 2018 val dataset in Table 3, which is a more challenging benchmark. Our INO reaches a new state-of-the-art which improves over DUL [1] by 1.3% in  $\mathcal{F}_m$  of the “seen” category and 0.6% in mean score. Compared with CRW [12] which is also trained on Kinetics, our INO outperforms by 1.4% in the mean score. As to MAST [15] which adopts 2 times larger resolution and a more advanced two-stage inference pipeline, i.e., detecting a ROI first and then considering the correspondence bounded by the ROI, our INO still outperforms by a significant margin (71.3% vs. 64.2%).

### 4.3 Qualitative Analysis

We visualize the mask propagation results in Fig. 3 and Fig. 4 for DAVIS-2017 val and YouTube-VOS 2018 val, respectively. For better comparison, we also illustrate the results for DUL [1] and CRW [12]. We observe that our INO achieves better qualitative results and is superior in the following aspects:

**Robust to Unseen Parts.** As illustrated in the first “motorcross-jump” scenario of Fig. 3, the right leg and back view of the rider is unseen in the given label of the first frame. The contrastive learning based methods CRW [12] and DUL [1] fail to track such unseen parts in the following sequences, while our INO tracks them successfully (see frame 15 and 35). We attribute this superiority mainly to the in-generative learning objective, which helps capture the complete semantic structure of the corrupted parts.

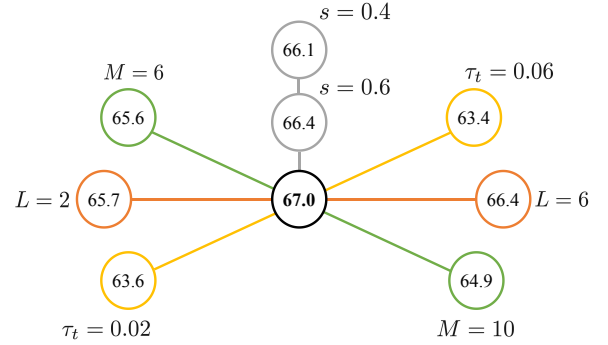
**Robust to Deformation.** Our INO shows better robustness to deformation compared with previous methods. For instance, in the “motorcross-jump” scenario of Fig. 3, the back wheel deforms significantly during the motion process, while only our INO can consistently identify the entire back wheel. The out-generative learning process fully exploits the semantic consistency between deformed objects from different frames during training. Intuitively, this supervised signal improves the robustness of features toward the deformation.

**Less Artifacts.** During the mask propagation process, the label may shift toward the background which shares a similar pattern with the target object, and this is termed as the “bleeding” artifacts in [1]. As illustrated in the first “motorcycle” case in Fig. 4, both CRW [12] and DUL [1] suffer this issue in frame 65 and 85, while our INO gives a more complete and stable mask even under a noisy background.

**Fine-grained Correspondence.** In the second “hat-trick” case of Fig. 4, we illustrate an extremely hard scenario that requires the

**Table 4: Effectiveness of In-N-Out generative learning objectives. All results are evaluated on DAVIS-2017 val benchmark.**

	Objective	$\mathcal{J}\&\mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{F}_m$	$\mathcal{F}_r$
Out	$\mathcal{L}_{out\_g2g}$	45.6	43.4	46.3	47.7	54.8
	$+\mathcal{L}_{out\_l2g}$	54.2	51.0	56.1	57.4	64.0
In	$+\mathcal{L}_{in\_mim}$	65.4	62.4	71.6	68.3	80.1
	$+\mathcal{L}_{in\_aff}$	<b>67.0</b>	<b>63.7</b>	<b>72.7</b>	<b>70.4</b>	<b>82.9</b>



**Figure 5: Ablation study of training parameters. We report the  $\mathcal{J}\&\mathcal{F}_m$  score evaluated on DAVIS-2017 val. Our baseline configuration (the centered one) is:  $M = 8, L = 4, \tau_t = 0.04, s = 0.8$ .**

ability to precisely capture the fine-grained details. Specifically, a small hat in high motion is thrown by the actor. As the label propagates, CRW [12] lost the hat gradually, and DUL [1] shifts to the actor’s face immediately, while only our INO tracks the hat consistently and precisely.

### 4.4 Ablation Studies

We investigate the effectiveness of learning objectives in Table 4, and the influence of training parameters in Fig. 5. The performance trained on Charades and evaluated on DAVIS-2017 val is reported.

**Effectiveness of In-N-Out Generative Learning.** In Table 4, we investigate the effectiveness of In-N-Out generative learning objectives. We set the out-generative learning between global crops as the baseline configuration and then add each term gradually. With only out-generative learning between global crops ( $\mathcal{L}_{out\_g2g}$ ), the model can achieve 45.6% in  $\mathcal{J}\&\mathcal{F}_m$ . After adding the out-generative learning between local and global crops ( $\mathcal{L}_{out\_l2g}$ ), the  $\mathcal{J}\&\mathcal{F}_m$  score boosts for 8.6%, which shows that the high-level semantic inference from local to global crops can significantly help to improve the performance. With out-generative learning only, the model achieves a performance of 54.2%. Notably, the performance is further improved significantly by 11.2% in  $\mathcal{J}\&\mathcal{F}_m$  score after including the in-generative learning via masked image modeling ( $\mathcal{L}_{in\_mim}$ ), which shows the importance of fine-grained structural semantic. Finally, the best performance is achieved after  $\mathcal{L}_{in\_aff}$  is introduced, which improves the temporal correspondence via bootstrapping.

**Temperature of Affinity Constraint.** In  $\mathcal{L}_{in\_aff}$ , we use  $\tau_s$  and  $\tau_t$  as the temperature for student and teacher affinity matrix, separately. We fix the student temperature  $\tau_s$  as 0.1 and change  $\tau_t$



**Table 5: Influence of different configurations of operating flipping and color jitter (F&C).**

	Global?	Local?	$\mathcal{J} \& \mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{F}_m$	$\mathcal{F}_r$
F&C	-	-	66.6	63.5	74.7	69.7	82.4
	✓	-	63.6	60.8	69.5	66.5	77.9
	-	✓	<b>67.0</b>	<b>63.7</b>	72.7	<b>70.4</b>	<b>82.9</b>
	✓	✓	66.2	63.3	73.3	69.1	82.1

as  $\{0.02, 0.04, 0.06\}$ , where smaller  $\tau_t$  gives sharper target distributions. As illustrated by the yellow one in Fig. 5,  $\tau_t = 0.04$  gives a moderate sharpness and performs best.

**Sequence Length.** We increase the sequence length  $L$  as  $\{2, 4, 6\}$ , and find that  $L = 4$  performs best, as illustrated by the orange one in Fig. 5. Longer sequence length brings larger difference between the paired frames, however, too much difference (e.g., the object may disappear in the later frames) incurs the mismatch between the source and target views, which increases the training noise. Therefore, a moderate sequence length ( $L = 4$ ) gives the best performance.

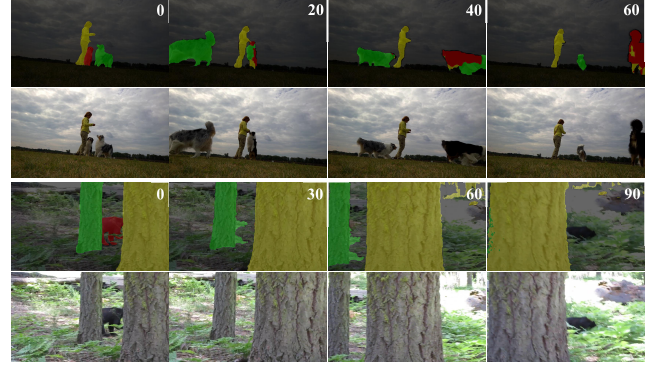
**Number of Local Crops.** We vary the local crops number  $M$  as  $\{6, 8, 10\}$ . More local crops bring more radical high-level semantic inference. As illustrated by the green one in Fig. 5,  $M = 8$  gives the best performance, which is tangibly better than  $M = 6$  and  $M = 10$ , e.g.,  $M = 10$  leads to a drop in  $\mathcal{J} \& \mathcal{F}_m$  score by 2.1%. We infer that too many local crops may lead to overfitting toward the noise, and a moderate  $M = 8$  is sufficient.

**Scale Threshold for Random Resized Cropping.** We use the random resized cropping which first samples views with scale range  $(0.05, s)$  for local crops and  $(s, 0.95)$  for global crops, and then resized them to  $64 \times 64$  and  $224 \times 224$ , respectively. We vary the threshold  $s$  as  $\{0.4, 0.6, 0.8\}$ . Larger  $s$  leads to more diverse local crops and more stable global crops containing richer information about the scene. By observing the gray one in Fig. 5, we find that the performance becomes better as  $s$  increases from 0.4 to 0.8. We infer that the global crops with rich global information are more helpful to the semantic inference from local to global.

**Flipping and Color Jitter.** After obtaining global and local crops via *random resized cropping*, we empirically perform *flipping* and *color jitter* (F&C) randomly only for local crops. We investigate 4 configurations in Table 5. We observe that operating F&C on global crops will reduce the performance. For instance, compared with not using F&C, the  $\mathcal{J} \& \mathcal{F}_m$  score drops by 3.0% from 66.6% to 63.6% after performing F&C on global crops. We infer that more stable global crops are preferred for the generation process. Performing F&C only on local crops provides the best performance on average (67.0% in  $\mathcal{J} \& \mathcal{F}_m$  score), therefore we take it as the default configuration.

#### 4.5 Limitation

We provide failure cases on DAVIS-2017 val and YouTube-VOS 2018 val in Fig. 6. We observe that the performance tends to be unsatisfactory under the following challenging scenarios: (i) *Multiple extremely similar instances*. For instance, the two dogs in case 1 of Fig. 6 are very similar in low-level vision, therefore the labels are confused during the later time steps. (ii) *Long-term and large-area occlusion*. As illustrated by case 2 of Fig. 6, the labels tend to lose the object after the long-term and large-area occlusion. We can observe

**Figure 6: Failure cases on DAVIS-2017 val (upper) and YouTube-VOS 2018 val (bottom).**

that there is still room for improvement in the aforementioned challenging cases.

## 5 CONCLUSION

In this paper, to tackle the challenging unsupervised learning for VOS, we proposed a simple yet effective framework called In-N-Out (INO) generative learning. The proposed INO is a novel fully-generative learning framework via in-view image recovery and out-view image imagination, which integrates both pixel-level and image-level optimization in a unified framework. Without any annotation data, we effectively learn robust visual representations in a self-supervised manner and achieve the new state-of-the-art performance among unsupervised learning methods for VOS. However, there is still a long way to go considering the analyzed limitations in § 4.5 and the performance gap between the unsupervised and the supervised methods. Thus, more effective algorithms are still required to alleviate this gap. We hope that our efforts will motivate more researchers and ease future research.

## ACKNOWLEDGMENTS

Xiao Pan and Zongxin Yang were in part supported by the Fundamental Research Funds for the Central Universities (No. 226-2022-00087).



## REFERENCES

- [1] Nikita Araslanov, Simone Schaub-Meyer, and Stefan Roth. 2021. Dense Unsupervised Learning for Video Segmentation. *Advances in Neural Information Processing Systems* 34 (2021).
- [2] Hangbo Bao, Li Dong, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254* (2021).
- [3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*. Springer, 213–229.
- [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9650–9660.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiuhua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).
- [7] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677* (2017).
- [8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems* 33 (2020), 21271–21284.
- [9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2021. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377* (2021).
- [10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
- [12] Allan Jabri, Andrew Owens, and Alexei Efros. 2020. Space-time correspondence as a contrastive random walk. *Advances in neural information processing systems* 33 (2020), 19545–19560.
- [13] Sangryul Jeon, Dongbo Min, Seungryong Kim, and Kwanghoon Sohn. 2021. Mining better samples for contrastive learning of temporal correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1034–1044.
- [14] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* (2017).
- [15] Zihang Lai, Erika Lu, and Weidi Xie. 2020. MAST: A memory-augmented self-supervised tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6479–6488.
- [16] Zihang Lai and Weidi Xie. 2019. Self-supervised learning for video correspondence flow. *arXiv preprint arXiv:1905.00875* (2019).
- [17] Chen Liang, Yu Wu, Tianfei Zhou, Wenguan Wang, Zongxin Yang, Yunchao Wei, and Yi Yang. 2021. Rethinking cross-modal interaction from a top-down perspective for referring video object segmentation. *arXiv preprint arXiv:2106.01061* (2021).
- [18] Qin Lin, Nuo Pang, and Zhiying Hong. 2021. Automated Multi-Modal Video Editing for Ads Video. In *Proceedings of the 29th ACM International Conference on Multimedia*. 4823–4827.
- [19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10012–10022.
- [20] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).
- [21] Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam. (2018).
- [22] Jiaxu Miao, Yunchao Wei, and Yi Yang. 2020. Memory aggregation networks for efficient interactive video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10366–10375.
- [23] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. 2018. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. In *The European Conference on Computer Vision (ECCV)*.
- [24] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. 2019. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9226–9235.
- [25] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. 2016. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 724–732.
- [26] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. 2017. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675* (2017).
- [27] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*. Springer, 510–526.
- [28] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*. PMLR, 10347–10357.
- [29] Jack Valmadre, Luca Bertinetto, Joao F Henriques, Ran Tao, Andrea Vedaldi, Arnold WM Smeulders, Philip HS Torr, and Efstratios Gavves. 2018. Long-term tracking in the wild: A benchmark. In *Proceedings of the European conference on computer vision (ECCV)*. 670–685.
- [30] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. 2018. Tracking emerges by colorizing videos. In *Proceedings of the European conference on computer vision (ECCV)*. 391–408.
- [31] Ning Wang, Wengang Zhou, and Houqiang Li. 2020. Contrastive transformation for self-supervised correspondence learning. *arXiv preprint arXiv:2012.05057* (2020).
- [32] Wenguan Wang, Tianfei Zhou, Fatih Porikli, David Crandall, and Luc Van Gool. 2021. A survey on deep learning technique for video segmentation. *arXiv preprint arXiv:2107.01153* (2021).
- [33] Xiaolong Wang, Allan Jabri, and Alexei A Efros. 2019. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2566–2576.
- [34] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. 2021. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8741–8750.
- [35] Jianghao Xiong, En-Lin Hsiang, Ziqian He, Tao Zhan, and Shin-Tson Wu. 2021. Augmented reality and virtual reality displays: emerging technologies and future perspectives. *Light: Science & Applications* 10, 1 (2021), 1–30.
- [36] Jiarui Xu and Xiaolong Wang. 2021. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10075–10085.
- [37] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. 2018. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 585–601.
- [38] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. 2019. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 879–888.
- [39] Zongxin Yang, Peike Li, Qianyu Feng, Yunchao Wei, and Yi Yang. 2019. Going deeper into embedding learning for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 0–0.
- [40] Zongxin Yang, Yunchao Wei, and Yi Yang. 2020. Collaborative video object segmentation by foreground-background integration. In *European Conference on Computer Vision*. Springer, 332–348.
- [41] Zongxin Yang, Yunchao Wei, and Yi Yang. 2021. Associating objects with transformers for video object segmentation. *Advances in Neural Information Processing Systems* 34 (2021), 2491–2502.
- [42] Zongxin Yang, Yunchao Wei, and Yi Yang. 2021. Collaborative video object segmentation by multi-scale foreground-background integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).
- [43] Zixu Zhao, Yueming Jin, and Pheng-Ann Heng. 2021. Modelling neighbor relation in joint space-time graph for video correspondence learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9960–9969.
- [44] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. 2021. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832* (2021).

## A ADDITIONAL IMPLEMENTATION DETAILS

We provide more training and inference details in this section. The code will be released upon acceptance.

### A.1 Training Details

After achieving global and local views via random resized cropping, we empirically perform random flip and random color jitter only for local crops, which leads to more stable global crops. Learnable position embedding is initialized with the resolution of  $64 \times 64$  and resized to the needed resolution (determined by the input resolution) with bicubic interpolation. The architecture of the projection head is the same as [4] and its layer number and output dimension are set as 3 and 4096, respectively. We train our network with adamw optimizer [21] and the weight decay follows the cosine schedule [20] from 0.04 to 0.4. The momentum of EMA is set as 0.996. We use the same video data loader as the released code of [12] and set the frameskip interval for each clip as 8. Considering the significant scale difference between these two datasets, we employ different learning rate schedules. Specifically, for Charades, we linearly warmup the learning rate for the first 5 epochs till the base value before decaying with a cosine schedule. The base value is determined by the linear scaling rule [7]:  $lr = 0.003 * \text{batchsize} * L / 1024$ . While for Kinetics-400, we reduce the linear warmup epochs to 2 and the learning rate is reduced to  $lr = 0.0003 * \text{batchsize} * L / 1024$  since it has significantly more iterations. By default, we train on 4 V100 (16GB) GPUs with a batchsize of 16 (4 clips per GPU) for Charades, and a batchsize of 32 on 8 V100 (16GB) GPUs for Kinetics-400.

### A.2 Inference Details

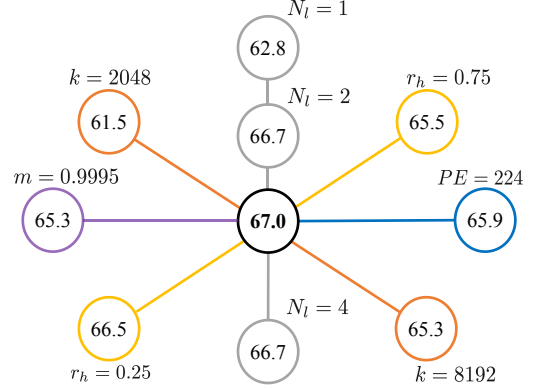
During inference, the given segmentation labels of the first frame are propagated toward the following frames via the label propagation algorithm. For fair comparison, we employ the same algorithm as previous methods [1, 4, 12, 36]. Specifically, given a  $d$ -dimensional feature embedding of the target pixel in the current frame, its cosine similarity *w.r.t* the first frame and the  $N_c$  previous context frames is calculated. For each frame, restrict attention [15], which means to only consider the pixels around the target pixel location with a radius of  $N_r$ , is applied. Then, for all the included reference pixels, the most similar  $N_k$  pixels are normalized with a temperature softmax. The final propagated label for the target pixel is the weighted combination of the reference pixel labels.

Same as previous works [1, 15, 36], the original image size is kept for inference. The label propagation is calculated via the patch tokens. The parameters of the label propagation algorithm on DAVIS-2017 val are  $N_k = 5$ ,  $N_c = 10$ ,  $N_r = 40$ . As to YouTube-VOS 2018 val, we adopt longer context ( $N_c = 20$ ) and larger radius ( $N_r = 50$ ) considering its longer video length in average and larger resolution (720p vs. 480p).

## B ADDITIONAL ABLATION STUDIES

We supplement more detailed ablation studies in this section. The performance trained on Charades and benchmarked on DAVIS-2017 val is reported.

### B.1 Ablation of Training Parameters



**Figure 7: Additional ablation studies of training parameters. We report the  $\mathcal{J} \& \mathcal{F}_m$  score evaluated on DAVIS-2017 val. Our baseline configuration (the centered one) is:  $m = 0.996, k = 4096, N_l = 3, r_h = 0.5, PE = 64$ .**

**B.1.1 Mask Ratio.** We perform masked image modeling at ratio  $r$ . Similar as [44], we randomly sample  $r$  from a uniform distribution from  $r_l$  to  $r_h$  at each iteration. We empirically fix  $r_l$  as 0.1 and vary  $r_h$  as  $\{0.25, 0.5, 0.75\}$ . As illustrated by the yellow one in Fig. 7,  $r_h = 0.5$  performs better than the rest ones. Larger  $r_h$  makes the recovery task harder, but may also include more noise. Therefore a moderate  $r_h = 0.5$  is sufficient.

**B.1.2 Position Embedding Resolution.** As in [4], the learnable position embedding (PE) is initialized with a fixed resolution, and interpolated to the target input resolutions. We tried two initial resolutions based on the used global and local crop size, *i.e.*,  $224 \times 224$  and  $64 \times 64$ . As shown by the blue one in Fig. 7, we find that  $64 \times 64$  gives better performance. This is reasonable considering that there are much more local crops (8 per frame) than global crops (1 per frame) during training. Initializing with  $64 \times 64$  resolution ensures that all the parameters in position embedding are fully optimized, while initializing with  $224 \times 224$  leads to the downsampling of position embedding in most cases (*i.e.*, when forwarding local crops), therefore the initialized parameters may be optimized in bias.

**B.1.3 EMA Momentum.** In the generic student-teacher framework, the teacher parameter is the Exponentially Moving Average (EMA) of the student parameters. We investigate the influence of the momentum  $m$  by varying it as  $\{0.996, 0.9995\}$ . As illustrated by the purple one in Fig. 7, we find that  $m = 0.996$  performs significantly better.

**B.1.4 Projection Head Layer Number.** We employ the same projection head architecture as [4], which is composed of a  $N_l$ -layer multi-layer perceptron (MLP) and a final fully connected layer with  $k$  dimensions output. We test the influence of the layer number  $N_l$  via changing it as  $\{1, 2, 3, 4\}$ . We empirically find that  $N_l = 1$  gives significantly lower performance, while  $N_l = 2, 3, 4$  performs similarly, and  $N_l = 3$  provides the best performance, as shown by the gray one in Fig. 7.

**B.1.5 Projection Head Output Dimension.** We investigate the influence of projection head output dimension  $k$  via varying it as  $\{2048, 4096, 8192\}$  (the orange one in Fig. 7). We observe that  $k = 4096$  performs best.

## B.2 Ablation of Backbone Architecture

**Table 6: Influence of backbone architectures. All results are evaluated on DAVIS-2017 val benchmark.**

Arch	$\mathcal{J} \& \mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{F}_m$	$\mathcal{F}_r$
ViT-T/8	65.7	62.8	<b>73.0</b>	68.7	81.6
ViT-S/8	<b>67.0</b>	<b>63.7</b>	72.7	<b>70.4</b>	<b>82.9</b>

We investigate the influence of backbones in Table 6. We additionally test ViT-T/8, which shares comparable parameters as ResNet-18 [11]. We observe that the performance can benefit from the increase of backbone parameters, *e.g.*, (67.0% vs. 65.7% in  $\mathcal{J} \& \mathcal{F}_m$  score).

## C ANALYSIS OF EFFICIENCY

**Table 7: Comparison of efficiency when training on Kinetics-400.**

Method	Arch	Param	GPUs	Time
VFS [36]	RN-50	23M	$8 \times 24GB$	1week
INO (ours)	ViT-S/8	21M	$8 \times 16GB$	1week

In Table 7, we compare the efficiency of our INO with the second best method VFS [36], which belongs to the image-level optimization using contrastive learning. As illustrated, with the similar training time, our INO employs a more light-weight backbone (21M vs. 24M) and requires less GPU memories ( $8 \times 16GB$  vs.  $8 \times 24GB$ ), while achieves significantly better performance, *e.g.*, 72.5% vs. 69.4% in  $\mathcal{J} \& \mathcal{F}_m$  score (see Table 2).

## D ADDITIONAL VISUALIZATION EXAMPLES

We provide more visualization examples of our INO on DAVIS-2017 val and YouTube-VOS 2018 val in Fig. 8 and Fig. 9, respectively.



Figure 8: Supplemented visualization examples of our INO on DAVIS-2017 val.





Figure 9: Supplemented visualization examples of our INO on YouTube-VOS 2018 val.