Unified Mask Embedding and Correspondence Learning for Self-Supervised Video Segmentation

Liulei Li[1,4*], Wenguan Wang[1†], Tianfei Zhou[2], Jianwu Li[3], Yi Yang[1]
[1] ReLER, CCAI, Zhejiang University [2] ETH Zurich [3] Beijing Institute of Technology [4] Baidu VIS

https://github.com/0liliulei/Mask-VOS

Abstract

The objective of this paper is self-supervised learning of video object segmentation. We develop a unified framework which simultaneously models cross-frame dense correspondence for locally discriminative feature learning and embeds object-level context for target-mask decoding. As a result, it is able to directly learn to perform mask-guided sequential segmentation from unlabeled videos, in contrast to previous efforts usually relying on an oblique solution — cheaply "copying" labels according to pixel-wise correlations. Concretely, our algorithm alternates between ***i)*** *clustering video pixels for creating pseudo segmentation labels ex nihilo; and* ***ii)*** *utilizing the pseudo labels to learn mask encoding and decoding for VOS. Unsupervised correspondence learning is further incorporated into this self-taught, mask embedding scheme, so as to ensure the generic nature of the learnt representation and avoid cluster degeneracy. Our algorithm sets state-of-the-arts on two standard benchmarks (*i.e.*, DAVIS$_{17}$ and YouTube-VOS), narrowing the gap between self- and fully-supervised VOS, in terms of both performance and network architecture design.*

1. Introduction

In this article, we focus on a classic computer vision task: accurately segmenting desired object(s) in a video sequence, where the target object(s) are defined by pixel-wise mask(s) in the first frame. This task is referred as (*one-shot*) *video object segmentation* (VOS) or *mask propagation* [1], playing a vital role in video editing and self-driving. Prevalent solutions [2–25] are built upon *fully supervised* learning techniques, costing intensive labeling efforts. In contrast, we aim to learn VOS from *unlabeled* videos — *self-supervised* VOS.

Due to the absence of mask annotation during training, existing studies typically degrade such self-supervised yet *mask-guided segmentation* task as a combo of *unsupervised correspondence learning* and correspondence based, *non-learnable mask warping* (*cf.* Fig. 1(a)). They first learn pixel-/patch-wise matching (*i.e.*, cross-frame correspondence) by exploring the inherent continuity in raw videos as free supervisory signals, in the form of **i)** a *photometric reconstruction* problem where each pixel in a target frame is desired to be recovered by *copying* relevant pixels in reference frame(s) [26–31]; **ii)** a *cycle-consistency* task that enforces matching of pixels/patches after forward-backward tracking [32–36]; and **iii)** a *contrastive matching* scheme that contrasts confident correspondences against unreliable ones [37–40]. Once trained, the dense matching model is used to approach VOS in a cheap way (Fig.1(a)): the label of a query pixel/patch is simply borrowed from previously segmented ones, according to their appearance similarity (correspondence score).

Though straightforward, these correspondence based "expedient" solutions come with two severe limitations: **First**,

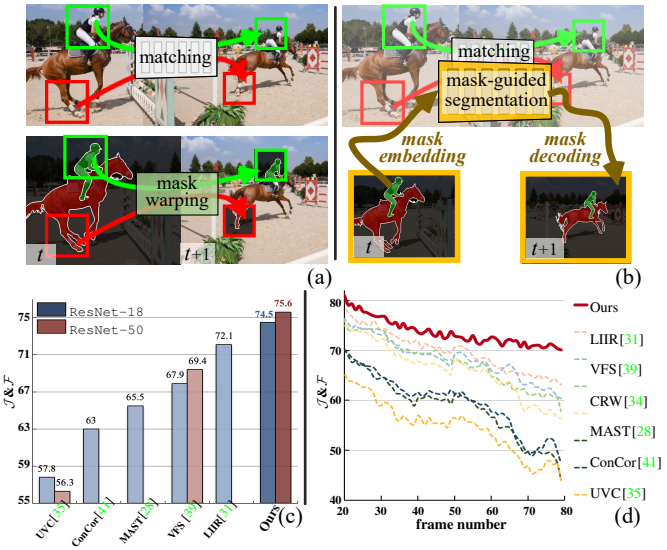

Figure 1. (a) Correspondence learning based self-supervised VOS, where mask tracking is simply degraded as correspondence matching mask warping. (b) We achieve self-supervised VOS by jointly learning mask embedding and correspondence matching. Our algorithm explicitly embeds masks for target object modeling, hence enabling mask-guided segmentation. (c) Performance comparison and (d) Performance over time, reported on DAVIS$_{17}$ [42] `val`.

*Work done during an internship at Baidu VIS.
†Corresponding author: *Wenguan Wang*.

they learn to match pixels instead of customizing VOS target – mask-guided segmentation, leaving a significant gap between the training goal and task/inference setup. During training, the model is optimized purely to discovery reliable, target-agnostic visual correlations, with no sense of object-mask information. Spontaneously, during testing/inference, the model struggles in employing first-/prior-frame masks to guide the prediction of succeeding frames. **Second**, from the view of mask-tracking, existing self-supervised solutions, in essence, adopt an obsolete, matching-/flow-based mask propagation strategy [43–47]. As discussed even before the deep learning era [48–50], such a strategy is sub-optimal. Specifically, without modeling the target objects, flow-based mask warping is sensitive to outliers, resulting in error accumulation over time [1]. Subject to the primitive matching-and-copy mechanism, even trivial errors are hard to be corrected, and often lead to much worse results caused by drifts or occlusions. This is also why current top-leading *fully supervised* VOS solutions [4, 5, 10–22] largely follow a *mask embedding learning* philosophy — embedding *frame-mask pairs*, instead of only frame images, into the segmentation network. With such explicit modeling of the target object, more robust and accurate mask-tracking can be achieved [1, 51].

Motivated by the aforementioned discussions, we integrate mask embedding learning and dense correspondence modeling into a compact, end-to-end framework for self-supervised VOS (*cf.* Fig. 1(b)). This allows us to inject the mask-tracking nature of the task into the very heart of our algorithm and model training. However, bringing the idea of mask embedding into self-supervised VOS is not trivial, due to the lack of mask annotation. We therefore achieve mask embedding learning in a *self-taught* manner. Concretely, our model is trained by alternating between **i)** space-time pixel clustering, and **ii)** mask-embedded segmentation learning. Pixel clustering is to automatically discover spatiotemporally coherent object(-like) regions from raw videos. By utilizing such pixel-level video partitions as pseudo ground-truths of target objects, our model can learn how to extract target-specific context from frame-mask pairs, and how to leverage such high-level context to predict the next-frame mask. At the same time, such self-taught mask embedding scheme is consolidated by self-supervised dense correspondence learning. This allows our model to learn transferable, locally discriminative representations by making full use of the spatiotemporal coherence in natural videos, and prevent the degenerate solution of the deterministic clustering.

Our approach owns a few distinctive features: **First**, it has the ability of directly learning to conduct mask-guided sequential segmentation; its training objective is completely aligned with the core nature of VOS. **Second**, by learning to embed object-masks into mask tracking, target-oriented context can be efficiently mined and explicitly leveraged for object modeling, rather than existing methods merely relying on local appearance correlations for label "copying". Hence our approach can reduce error accumulation (*cf.* Fig. 1(d)) and perform more robust when the latent correspondences are ambiguous, *e.g.*, deformation, occlusion or one-to-many matches. **Third**, our mask embedding strategy endows our self-supervised framework with the potential of being empowered by more advanced VOS model designs developed in the fully-supervised learning setting.

Through embracing the powerful idea of mask embedding learning as well as inheriting the merits of correspondence learning, our approach favorably outperforms state-of-the-art competitors, *i.e.*, **3.2**%, **2.5**%, and **2.2**% mIoU gains on DAVIS$_{17}$ [42] `val`, DAVIS$_{17}$ `test-dev` and YouTube-VOS [52] `val`, respectively. In addition to narrowing the performance gap between self- and fully-supervised VOS, our approach establishes a tight coupling between them in the aspect of model design. We expect this work can foster the mutual collaboration between these two relevant fields.

2. Related Work

Fully Supervised Learning for VOS. Given a target object outlined in the first frame, VOS aims to precisely extract this object from the rest frames. Fully supervised deep learning based solutions have became the mainstream in this field, and can be broadly grouped into three categories [1]: *online finetuning* based [15, 16] (*i.e.*, training a segmentation network separately on each test-time given object), *propagation* based [12, 13, 17] (*i.e.*, using the latest mask to infer the upcoming frame mask), and *matching* based [4, 5, 10, 11, 18, 20] (*i.e.*, classifying pixels according to their similarities to the target object). Despite assorted motivations and technique details, almost all the top-leading approaches are built upon a common principle – embedding paired frame and mask, *e.g.*, (I_r, Y_r), into the segmentation network $\mathcal{S}$:

$$Y_q = \mathcal{S}(I_q, \{\mathcal{V}(I_r, Y_r)\}_r), \tag{1}$$

where Y_q is the mask predicted for a given query frame I_q; the function $\mathcal{V}$ learns to deduce target-specific context from the reference (I_r, Y_r); $\{(I_r, Y_r)\}_r$ can be the initial frame-mask pair [4, 5, 11, 17, 18], and/or paired historical frames and mask predictions [6, 9, 10, 20, 53–55]. Then the target-aware context is used to guide the segmentation (mask decoding) of the query I_q. For instance, [4, 5, 11, 18, 56] store foreground and background representations and classify pixels by nearest neighbor retrieval or feature decoding. Some others directly project reference frame and mask into a joint (space-time) embedding space, which is subsequently used for mask propagation [17] or feature matching [6, 9, 10, 20, 54, 55]. A few recent methods [15, 57] treat mask-derived object representation as the target of a few-shot learner [15] or a prior for joint inductive and transductive learning [57].

Inspired by these achievements, for the first time, we exploit the idea of mask embedding in the self-supervised VOS

setting. We achieve this through automatic space-time clustering and using deterministic cluster assignments as pseudo groundtruths to supervise the learning of mask embedding and decoding, without the aid of manual labels. In this way, our self-supervised model is capable of explicitly and comprehensively describing the target object, hence achieving more robust and accurate, target-oriented segmentation.

Self-supervised Learning for VOS. Learning VOS in a self-supervised manner is appealing, as it eliminates the heavy annotation budget required by the fully supervised algorithms. Due to the absence of mask annotation, existing self-supervised methods take an *expedient* solution: they learn to find correspondence between two frames, instead of learning mask-guided segmentation. During inference, the first-frame mask is directly copied to the rest frames based on cross-frame correspondence. Specifically, given two frames I_r and I_q, their dense representations $\boldsymbol{I}_r, \boldsymbol{I}_q \in \mathbb{R}^{HW\times D}$ are first extracted by a shallow neural encoder $\mathcal{E}$ (typically `ResNet-18` [58]), and their pairwise affinity matrix can be computed as:

$$A_r^q = \texttt{softmax}(\boldsymbol{I}_r \boldsymbol{I}_q^\top) \ \in [0,1]^{HW\times HW}, \tag{2}$$

where `softmax` is row-wise. The resultant affinity A_r^q gives the strength of all the pixel pairwise correspondence between $\boldsymbol{I}_r$ and $\boldsymbol{I}_q$. One main benefit is that, once $\mathcal{E}$ is trained, it can be used to estimate cross-frame correspondence; then VOS is approached by warping the mask Y_r of a reference frame I_r to the query frame I_q based on: $Y_q = A_r^{q\top} Y_r$. Thus the central problem is to design a surrogate task to supervise $\mathcal{E}$ to estimate reliable intra-frame affinity A_r^q. Basically, three types of surrogate tasks are developed, yet all exploit the correlation among frames: **i)** *Photometric reconstruction* [15, 26–28, 37, 38, 41]. Here the affinity matrix A_r^q is estimated to reconstruct the query frame: $\tilde{I}_q = A_r^{q\top} I_r$, invoking a photometric reconstruction objective: $\mathcal{L}_{\text{Re}} = \|I_q - \tilde{I}_q\|^2$; **ii)** *Cycle-consistency tracking* [32–35, 59–61]. The affinity A_r^q is used to guide a cycle of forward and backward tracking, leading to a cycle-consistency loss: $\mathcal{L}_{\text{Cyc}} = \|A_r^q A_q^r - \mathbb{1}\|^2$, where $\mathbb{1}$ is an identity matrix with proper size; and **iii)** *Contrastive matching* [37–39]. Drawn inspiration from unsupervised contrastive learning [62, 63], temporal correspondence learning is achieved by contrasting the affinity between positive, matched pixel pairs (i, i^+), against the affinity between negative, unrelated ones (i, i^-): $\mathcal{L}_{\text{Con}} = -\log(\exp(A(i,i^+))/\sum_{i^-}\exp(A(i,i^-)))$. The positive pairs are often pre-defined as spatiotemporally adjacent pixels, so as to capture the coherence residing videos [37, 38], while [39] shows that fine-grained correspondence can be captured by directly contrasting frame samples. For long-range matching, multiple reference frames are considered in practice [28, 34, 37, 38].

Our algorithm is fundamentally different from existing self-supervised VOS solutions. Through self-taught mask embedding learning, our method begets mask-guided segmentation. Thus the nature of VOS is captured by our network architecture and training, rather than existing methods treating the task as a derivant of unsupervised correspondence learning. Further, our method is principled; correspondence learning can be seamlessly incorporated for regularizing representation learning, while the concomitant shortcomings, *e.g.*, no sense of target-specific information, error accumulation over time, and misalignment between training and inference modes, are naturally alleviated.

3. Methodology

At a high level, our self-supervised VOS solution jointly learns mask embedding and visual correspondence from raw videos. It absorbs the powerful idea of mask-embedded segmentation (*cf.* Eq. 1) in fully supervised VOS; meanwhile, it inherits the merits of existing unsupervised correspondence based regime (*cf.* Eq. 2) in learning generic, dense features. As a result, our solution can be formulated as (*cf.* Fig. 2):

$$Y_q = \mathcal{D}\big(\mathcal{E}(I_q), \{\mathcal{V}([I_{r_n}, Y_{r_n}])\}_n\big) \tag{3}$$

self-supervised dense correspondence learning §3.2 — self-supervised mask embedding learning §3.1

where $[\,\cdot\,]$ stands for concatenation. Basically, our model utilizes a set of reference frame-mask pairs, *i.e.*, $(I_{r_n}, Y_{r_n})_n$, to predict/decode the mask of each query frame I_q, learnt in a self-supervised manner. Our model has three core parts:

- ***Visual Encoder*** $\mathcal{E}$, which maps each query frame I_q into a dense representation tensor: $\boldsymbol{I}_q = \mathcal{E}(I_q) \in \mathbb{R}^{HW\times D}$. We instantiate $\mathcal{E}$ as `ResNet-18` or `ResNet-50`.
- ***Frame-Mask Encoder*** $\mathcal{V}$ for mask embedding. It takes a pair of a reference frame I_r and corresponding mask Y_r as inputs, and extracts target-specific context, *i.e.*, $\boldsymbol{V}_r = \mathcal{V}([I_r, Y_r]) \in \mathbb{R}^{HW\times D'}$, to guide the segmentation/mask decoding of I_q. $\mathcal{V}$ has a similar network architecture with $\mathcal{E}$, but the input and output dimensionality are different and the network weights are unshared.
- ***Mask Decoder*** $\mathcal{D}$, which is a small CNN for mask decoding. With the help of target-rich context $\{\boldsymbol{V}_{r_n}\}_n$ collected from a set of reference frame-mask pairs $\{(I_{r_n}, Y_{r_n})\}_n$, $\mathcal{D}$ makes robust prediction, *i.e.*, Y_q, for the query frame I_q.

As for training, to mitigate the dilemma caused by the absence of true labels of $\{Y_{r_n}\}_n$ and Y_q, we conduct unsupervised space-time pixel clustering for automatic mask creation and train the whole network, including $\mathcal{E}$, $\mathcal{V}$, and $\mathcal{D}$, for mask embedding and decoding (§3.1). Moreover, unsupervised contrastive correspondence learning (§3.2) is introduced to boost dense visual representation learning of $\mathcal{E}$.

3.1. Self-supervised Mask Embedding Learning

For self-supervised mask embedding learning, we alternatively perform two steps: **Step1**: clustering of video pixels on the visual feature space $\mathcal{E}$ so as to generate spatiotemporally compact segments; and **Step 2**: the space-time cluster assignments serve as pseudo masks to supervise our whole

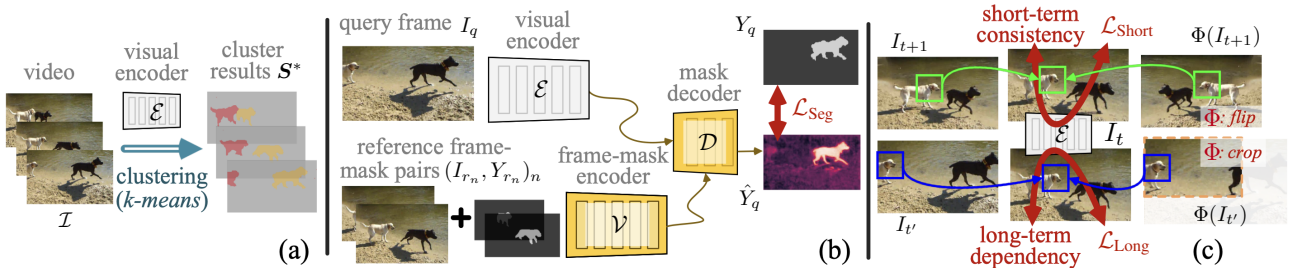

Figure 2. Our self-supervised VOS framework: **(a-b)** space-time pixel clustering based mask embedding learning (§3.1) for the whole network (including $\mathcal{E}$, $\mathcal{V}$, and $\mathcal{D}$), and **(c)** short- and long-term correspondence learning (§3.2) for the visual encoder $\mathcal{E}$ only.

network (including $\mathcal{E}, \mathcal{V}$, and $\mathcal{D}$), which learns VOS as mask-embedded sequential segmentation. After that, the improved visual representation $\mathcal{E}$ will in turn facilitate clustering.

Step 1: Space-time Clustering. The goal of this step is to partition each training video $\mathcal{I}$ into M space-time consistent segments (see Fig. 2(a)). For each pixel $i \in \mathcal{I}$, let $\boldsymbol{i} \in \mathbb{R}^D$ denote its visual embedding (extracted from the visual encoder $\mathcal{E}$), and $\boldsymbol{s}_i \in \{0,1\}^M$ its one-hot cluster assignment vector. Clustering of all the pixels in $\mathcal{I}$ into M clusters can be achieved by solving the following optimization problem:

$$\min_{\boldsymbol{C},\boldsymbol{S}} \sum\nolimits_{i\in\mathcal{I}} \|\boldsymbol{i} - \boldsymbol{C}\boldsymbol{s}_i\|, \quad s.t.\ \ \boldsymbol{s}_i \in \{0,1\}^M, \quad \mathbf{1}^\top \boldsymbol{s}_i = 1. \quad (4)$$

Here $\boldsymbol{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_M] \in \mathbb{R}^{D\times M}$ is the cluster centroid matrix, where $\boldsymbol{c}_m \in \mathbb{R}^D$ refers to the centroid of m-*th* cluster, and $\boldsymbol{S} = [\boldsymbol{s}_i]_i$ stores the cluster assignments of all the pixels in $\mathcal{I}$. $\mathbf{1}$ is a M-dimensional all-one vector. While many clustering methods have been designed to solve Eq. 4, for simplicity, we use the most classic one – k-means, which finds the optimal $\boldsymbol{C}^*$ and $\boldsymbol{S}^*$ in an EM fashion. Moreover, to pursue spatiotemporally compact clusters, for each pixel $i \in \mathcal{I}$, we supply its visual embedding $\boldsymbol{i}$ with a 3D sinusoidal position encoding vector [64, 65]. In practice, only a small number of EM steps (*i.e.*, 100) can deliver satisfactory clustering results, taking about 2 seconds per video, averaged on our training dataset – YouTube-VOS [52].

Step 2: Mask-embedded Segmentation Learning. In this step, our model utilizes clustering results as pseudo ground-truths (see Fig. 2(b)), to directly learn VOS as mask embedding and decoding. For each training video $\mathcal{I}$, we sample $N+1$ frames $\{I_{r_1}, I_{r_2}, \cdots, I_{r_N}, I_q\}$ and their masks $\{Y_{r_1}, Y_{r_2}, \cdots, Y_{r_N}, Y_q\}$, as training examples. The pseudo masks are naturally derived from the assignment matrix $\boldsymbol{S}^*$, corresponding to the pixel-level assignment of a certain cluster. The training examples are used to teach our model to refer to the first N frame-mask pairs $\{(I_{r_n}, Y_{r_n})\}_n$ to segment the last query frame I_q — predicting Y_q. As such, our model can learn i) *mask embedding*: how to extract target-specific context from $\{(I_{r_n}, Y_{r_n})\}_n$; and ii) *mask decoding*: how to make use of target-specific context to segment the target in I_q.

More specifically, we first respectively apply our visual encoder $\mathcal{E}$ and frame-mask encoder $\mathcal{V}$ over each reference frame I_{r_n} and each reference frame-mask pair (I_{r_n}, Y_{r_n}), to obtain visual and target-specific embeddings:

$$\boldsymbol{I}_{r_n} = \mathcal{E}(I_{r_n}) \in \mathbb{R}^{HW\times D}, \quad \boldsymbol{V}_{r_n} = \mathcal{V}([I_{r_n}, Y_{r_n}]) \in \mathbb{R}^{HW\times D'}. \quad (5)$$

We respectively stack all the reference visual and target-specific embeddings: $\boldsymbol{I}_r = [\boldsymbol{I}_{r_1}, \cdots, \boldsymbol{I}_{r_N}] \in \mathbb{R}^{NHW\times D}$, and $\boldsymbol{V}_r = [\boldsymbol{V}_{r_1}, \cdots, \boldsymbol{V}_{r_N}] \in \mathbb{R}^{NHW\times D'}$. To leverage $\boldsymbol{V}_r$ to boost the prediction of I_q, we need to mine useful context, related to I_q, from $\boldsymbol{V}_r$. Given the visual embedding $\boldsymbol{I}_q \in \mathbb{R}^{HW\times D}$ of I_q (extracted from $\mathcal{E}$), we estimate the affinity between the query I_q and reference frames $\{I_{r_n}\}_n$ (analogous to Eq. 2):

$$A = \texttt{softmax}(\boldsymbol{I}_r \boldsymbol{I}_q^\top) \ \in \mathbb{R}^{NHW\times HW}. \quad (6)$$

Hence target-specific, supportive features are accordingly assembled to yield:

$$\boldsymbol{V}_q = A^\top \boldsymbol{V}_r \ \in \mathbb{R}^{HW\times D'}. \quad (7)$$

Here $\boldsymbol{V}_q$ absorbs existent object observations in the reference set $\{(I_{r_n}, Y_{r_n})\}_n$, revealing for I_q whether each pixel thereof belongs to the target object or not. Given precise segmentation groundtruths, it is relatively easy for fully supervised methods [9, 10, 54] to learn to directly decode $\boldsymbol{V}_q$ into segmentation mask. However, this strategy does not work well in our case since the pseudo labels are inevitably noisy and less accurate, compared with the real groundtruths. To tackle this, we achieve mask decoding through a *mask refinement* scheme, which makes more explicit use of reference masks. Specifically, we first construct a coarse mask $\bar{Y}_q$ for I_q by warping the reference masks $\{Y_{r_n}\}_n$ *w.r.t.* the affinity A:

$$\bar{Y}_q = A^\top [Y_{r_1}, Y_{r_2}, \cdots, Y_{r_n}] \ \in \mathbb{R}^{HW}. \quad (8)$$

The segmentation prediction $\hat{Y}_q$ for the query I_q is made as:

$$\hat{Y}_q = \mathcal{D}([\boldsymbol{V}_q, \bar{\boldsymbol{V}}_q]), \quad \bar{\boldsymbol{V}}_q = \mathcal{V}([I_q, \bar{Y}_q]) \ \in \mathbb{R}^{HW\times D'}. \quad (9)$$

Here the frame-mask encoder $\mathcal{V}$ (*cf.* Eq. 5) is smartly revoked to get another target-specific embedding $\bar{\boldsymbol{V}}_q$, from the pair of the query frame I_q and warped coarse mask $\bar{Y}_q$. This also elegantly resembles the mask copying strategy adopted in existing correspondence-based self-supervised VOS models. Conditioned on the concatenation of $\boldsymbol{V}_q$ and $\bar{\boldsymbol{V}}_q$, the mask decoder $\mathcal{D}$ outputs a finer mask $\hat{Y}_q$. In practice we find our mask refinement strategy can ease training and bring better performance (related experiments can be found in Table 4e).

Given the pseudo segmentation label Y_q and prediction $\hat{Y}_q$ of I_q, our whole model is supervised by minimizing the standard cross-entropy loss $\mathcal{L}_{\text{CE}}$:

$$\mathcal{L}_{\text{Seg}} = \sum\nolimits_{\mathcal{I}} \mathcal{L}_{\text{CE}}(\hat{Y}_q, Y_q). \tag{10}$$

3.2. Self-supervised Dense Correspondence Learning

An appealing aspect of our mask embedding framework is that it is general enough to naturally incorporate unsupervised correspondence learning to specifically reinforce visual representation $\mathcal{E}$. This comes with a few advantages: First, this allows our model to exploit the inherent coherence in natural videos as free supervisory signals to promote the transferability and sharpen the discriminativeness of $\mathcal{E}$. Second, correspondence learning provides initial meaningful features for clustering (*cf.* Eq. 4), which is prone to degeneracy (*i.e.*, allocating most samples to the same cluster) caused by poor initialization [66]. Third, our segmentation model involves the computation of intra-frame affinity A (*cf.* Eqs. 5-7), raising a strong demand for efficiently modeling dense correspondence within our framework. Along with recent work of contrastive matching based correspondence learning [37–39], we comprehensively explore intrinsic continuity within raw videos in both *short-term* and *long-term* time scales, to boost the learning of $\mathcal{E}$ (see Fig. 2(c)).

Short-term Appearance Consistency. Temporally adjacent frames typically exhibit continuous and trivial appearance changes [59, 67]. To accommodate this property, we enforce *transformation-equivariance* [68–71] between our adjacent frame representations. Given two **successive** frames $I_t, I_{t+1} \in \mathcal{I}$, their representations, delivered by $\mathcal{E}$, are constrained to be **equivariant** against geometric transformations (*i.e.*, scaling, flipping, and cropping). Specifically, denote Φ as a random transformation, our *equivariance based short-term appearance consistency constraint* can be expressed as:

$$\left.\begin{array}{l} \text{❶ } \mathcal{E}(I_t) \approx \mathcal{E}(I_{t+1}) \\ \quad \text{short-term consistency} \\ \text{❷ } \mathcal{E}(\Phi(I_t)) = \Phi(\mathcal{E}(I_t)) \\ \quad \text{transformation-equivariance} \end{array}\right\} \Rightarrow \mathcal{E}(\Phi(I_t)) \approx \Phi(\mathcal{E}(I_{t+1})) \text{ ❸}. \tag{11}$$

Here ❶ states the short-term consistency property; ❷ refers to the equivariance constraint on a single image [71], *i.e.*, an imagery transformation Φ of I_t should lead to a correspondingly transformed feature [38]. By bringing ❷ into ❶, we prevent trivial solution, *i.e.*, $\mathcal{E}(I_t) \equiv \mathcal{E}(I_{t+1})$, when directly optimizing $\mathcal{E}$ via ❶, and eventually get ❸.

Following ❸, we first get the feature of transformed I_t: $\boldsymbol{X}'_t = \mathcal{E}(\Phi(I_t)) \in \mathbb{R}^{HW \times D}$, and transformed feature of I_{t+1}: $\boldsymbol{X}_{t+1} = \Phi(\mathcal{E}(I_{t+1})) \in \mathbb{R}^{HW \times D}$. Denote *k-th* pixel feature of $\boldsymbol{X}_{t+1}$ (resp. $\boldsymbol{X}'_t$) as $\boldsymbol{x}^k_{t+1} \in \mathbb{R}^D$ (resp. $\boldsymbol{x}'^k_t \in \mathbb{R}^D$)[1], our short-term consistency loss is computed as:

$$\mathcal{L}_{\text{Short}} = -\sum\nolimits_{\mathcal{I}} \sum\nolimits_k \log \frac{\exp(\langle \boldsymbol{x}^{k\top}_{t+1} \boldsymbol{x}'^k_t \rangle)}{\sum_l \exp(\langle \boldsymbol{x}^{k\top}_{t+1} \boldsymbol{x}'^l_t \rangle)}, \tag{12}$$

where $\langle \boldsymbol{x}^{k\top}_{t+1} \boldsymbol{x}'^l_t \rangle$ gives cosine similarity based affinity between *k-th* pixel feature of $\boldsymbol{X}_{t+1}$ and *l-th* pixel feature of $\boldsymbol{X}'_t$. Eq. 12 captures local appearance continuity by contrasting affinity between aligned pixel feature pairs, *i.e.*, $\boldsymbol{x}^k_{t+1}$and $\boldsymbol{x}'^k_t$ against non-corresponding ones, *i.e.*, $\boldsymbol{x}^k_{t+1}$ and $\{\boldsymbol{x}'^l_t\}_{l \neq k}$, with an extra transformation equivariance based constraint.

Long-term Semantic Dependency. In addition to considering the local consistency among adjacent frames, we exploit long-term coherence of visual content among distant frames [72, 73]. To address this property, we enforce transformation equivariance between representations of *arbitrary* frame pairs (sampled from the same video) after *alignment*. Given two **distant** frames $I_t, I_{t'} \in \mathcal{I}$ (*s.t.* $|t - t'| \geq 5$), their representations, after being **aligned** *w.r.t.* their affinity $A^t_{t'}$, are constrained to be **equivariant** against geometric transformations. In particular, denote $A^t_{t'} \in [0,1]^{HW \times HW}$ (resp. $A^t_{\Phi(t')} \in [0,1]^{HW \times HW}$) as the affinity between I_t and $I_{t'}$ (resp. I_t and $\Phi(I_{t'})$), our *equivariance based long-term semantic dependency constraint* can be expressed as:

$$\left.\begin{array}{l} \text{❹ } \mathcal{E}(I_t) \approx A^{t\top}_{t'} \mathcal{E}(I_{t'}) \\ \quad \text{long-term dependency} \\ \text{❷ } \mathcal{E}(\Phi(I_t)) = \Phi(\mathcal{E}(I_t)) \\ \quad \text{transformation-equivariance} \end{array}\right\} \Rightarrow \mathcal{E}(I_t) \approx A^{t\top}_{\Phi(t')} \Phi(\mathcal{E}(I_{t'})) \text{ ❺}. \tag{13}$$

Here ❹ states the long-term dependency property; ❷ poses the equivariance constraint, as in Eq. 11. By bringing ❷ into ❹, we prevent trivial solution, *i.e.*, $\mathcal{E}(I_t) \equiv \mathcal{E}(I_{t'})$, when directly optimizing $\mathcal{E}$ via ❹, and eventually get ❺. Specifically, similar to ❹, we have $\mathcal{E}(I_t) \approx A^{t\top}_{\Phi(t')} \mathcal{E}(\Phi(I_{t'}))$; then with ❷, we obtain $\mathcal{E}(I_t) \approx A^{t\top}_{\Phi(t')} \mathcal{E}(\Phi(I_{t'})) = A^{t\top}_{\Phi(t')} \Phi(\mathcal{E}(I_{t'}))$.

Following ❺, we get the feature of transformed $I_{t'}$: $\boldsymbol{X}'_{t'} = \mathcal{E}(\Phi(I_{t'})) \in \mathbb{R}^{HW \times D}$, transformed feature of $I_{t'}$: $\boldsymbol{X}_{t'} = \Phi(\mathcal{E}(I_{t'})) \in \mathbb{R}^{HW \times D}$, and the original feature of I_t: $\boldsymbol{I}_t = \mathcal{E}(I_t) \in \mathbb{R}^{HW \times D}$. For *k-th* pixel (feature) of $\boldsymbol{X}'_{t'}$, we first find the matching (*i.e.*, the most similar) pixel o_k in $\boldsymbol{I}_t$ as:

$$o_k = \underset{o \in \{1, \cdots, HW\}}{\arg\max}\ a_{k,o}, \qquad a_{k,o} = \frac{\exp(\langle \boldsymbol{x}'^{k\top}_{t'} \boldsymbol{i}^o_t \rangle)}{\sum_l \exp(\langle \boldsymbol{x}'^{k\top}_{t'} \boldsymbol{i}^l_t \rangle)}, \tag{14}$$

where $\boldsymbol{i}^o_t \in \mathbb{R}^D$ refers to *o-th* pixel feature of $\boldsymbol{I}_t$, and $a_{k,o}$ corresponds to (k,o)-*th* element of the affinity $A^t_{\Phi(t')}$ between $\Phi(I_{t'})$ and I_t. Then, the dominant index ok serves as pseudo labels for our temporally-distant matching and our long-term dependency loss is computed as:

$$\mathcal{L}_{\text{Long}} = -\sum\nolimits_{\mathcal{I}} \sum\nolimits_k \log \frac{\exp(\langle \boldsymbol{x}^{k\top}_{t'} \boldsymbol{i}^{o_k}_t \rangle)}{\sum_l \exp(\langle \boldsymbol{x}^{k\top}_{t'} \boldsymbol{i}^l_t \rangle)}. \tag{15}$$

Eq. 15 addresses global semantic dependencies by contrasting affinity between aligned pixel feature pairs, *i.e.*, $\boldsymbol{x}^k_{t'}$ and $\boldsymbol{i}^{o_k}_t$, against non-corresponding ones, *i.e.*, $\boldsymbol{x}^k_{t'}$ and $\{\boldsymbol{i}^l_t\}_{l \neq o_k}$, under an equivariant representation learning scheme.

[1] For clarity, the symbols for frame and pixel features in §3.2 are slightly redefined as $\boldsymbol{X}$ and $\boldsymbol{x}$, instead of using $\boldsymbol{I}$ and $\boldsymbol{i}$ as in §3.1.

3.3. Implementation Details

Full Loss. The overall training loss is:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{Seg}} + \mathcal{L}_{\text{Corr}} \\ &= \mathcal{L}_{\text{Seg}} + \lambda_1 \mathcal{L}_{\text{Short}} + \lambda_2 \mathcal{L}_{\text{Long}},\end{aligned} \tag{16}$$

where the coefficients are empirically set as: $\lambda_1 = 0.1$ and $\lambda_2 = 0.5$.

Network Configuration. For the *visual encoder* $\mathcal{E}$, we instantiate it as `ResNet-18` or `ResNet-50` in our experiments. For `ResNet-18`, the spatial strides of the second and last residual blocks are removed to yield an output stride of 8, as in [34, 38, 41]. For `ResNet-50`, we follow [32] to take features from `res4`, and drop its stride to preserve more spatial details. For the *frame-mask encoder* $\mathcal{V}$, it has a similar structure as $\mathcal{E}$, expect for the input and output dimensionality. On the top of $\mathcal{E}$ and $\mathcal{V}$, two 1×1 convolution layers are separately added to reduce the output dimensions of $\mathcal{E}$ and $\mathcal{V}$ to $D = 128$ and $D' = 512$, respectively. For the *mask decoder* $\mathcal{D}$, it consists of two Residual blocks that are connected with $\mathcal{E}$ through skip layers, and a 1×1 convolution layer to produce the final segmentation prediction.

Training. We follow [74] to pre-train the backbone network $\mathcal{E}$ on YouTube-VOS for 300 epochs, enabling reliable initial clustering. Then, we conduct the main training for a total of 400 epochs using Adam optimizer with batch size 16 and base learning rate 1e-4, on one Tesla A100 GPU. In the first 300 epochs, the whole network is trained with only the correspondence loss $\mathcal{L}_{\text{Corr}}$. The learning rate is scheduled following a "step" policy, decayed by multi-plying 0.5 every 100 epochs. In the last 100 epochs, the whole network is trained using the full loss $\mathcal{L}$, with fixed learning rate 1e-5. The first time-space clustering is made at epoch 300 for creating initial pseudo segmentation labels. Afterwards, the pseudo labels are updated by conducting re-clustering at every 10 epochs. During clustering, we abandon over-size clusters, *i.e.*, accounting for more than 40% of video pixels. These big clusters are typically scene background, like sky and grass; only the remaining pixel clusters/segments are used as pseudo labels. Random scaling, cropping, and flipping are used for data augmentation, and the training image size is set to 256×256. In each mini-batch, we sample 3 frames per video, and adopt the strategy in [9, 10] to learn mask decoding with two reference frames (*i.e.*, $N = 2$).

Testing. Once trained, our model is applied to test videos without any fine-tuning. Following [34, 38], for each query frame, we take the first frame (providing reliable object mask information), and, if applicable, its prior 20 frames (capturing diverse object patterns), as well as their masks, as reference for segmentation prediction. In addition, we repeatedly feed the prediction $\hat{Y}_q$ back to the mask decoder $\mathcal{D}$ for iterative refinement. We find this strategy brings better results while requiring no extra parameters, with only marginal sacrifice of inference speed (see Table 4e).

Method	Backbone	Dataset(size)	$\mathcal{J}\&\mathcal{F}_m\uparrow$	$\mathcal{J}_m\uparrow$	$\mathcal{J}_r\uparrow$	$\mathcal{F}_m\uparrow$	$\mathcal{F}_r\uparrow$
Colorization[26][ECCV18]	`ResNet-18`	Kinetics(- , 800 hours)	34.0	34.6	34.1	32.7	26.8
CorrFlow[27][BMVC19]	`ResNet-18`	OxUvA(- , 14 hours)	50.3	48.4	53.2	52.2	56.0
TimeCycle[32][CVPR19]	`ResNet-50`	VLOG(- , 344 hours)	48.7	46.4	50.0	50.0	48.0
UVC[35][NeurIPS19]	`ResNet-18`	C+Kinetics(30K, 800 hours)	57.8	56.3	65.0	59.2	64.1
MuG[59][CVPR20]	`ResNet-18`	OxUvA(- , 14 hours)	54.3	52.6	57.4	56.1	58.1
MAST[28][CVPR20]	`ResNet-18`	Youtube-VOS(- , 5.58 hours)	65.5	63.3	73.2	67.6	77.7
CRW[34][NeurIPS20]	`ResNet-18`	Kinetics(- , 800 hours)	68.3	65.5	78.6	71.0	82.9
ConCorr[41][AAAI21]	`ResNet-18`	C+TrackingNet(30K, 300 hours)	63.0	60.5	70.6	65.5	73.0
CLTC[37][CVPR21]	`ResNet-18`	Youtube-VOS(- , 5.58 hours)	70.3	67.9	78.2	72.6	83.7
JSTG[60][ICCV21]	`ResNet-18`	Kinetics(- , 800 hours)	68.7	65.8	77.7	71.6	84.3
VFS [39][ICCV21]	`ResNet-18`	Kinetics(- , 800 hours)	67.9	65.0	77.2	70.8	82.3
	`ResNet-50`		69.4	66.7	78.6	72.0	85.2
DINO[74][ICCV21]	`ResNet-50`	I(1.28M, -)	56.2	54.5	58.1	57.9	60.3
	`ViT-B/8`		71.4	67.9	81.6	74.9	85.4
DUL[38][NeurIPS21]	`ResNet-18`	Youtube-VOS(- , 5.58 hours)	69.3	67.1	81.2	71.6	84.9
SCR[40][CVPR22]	`ResNet-18`	Kinetics(- , 800 hours)	70.5	67.4	78.8	73.6	84.6
LIIR[31][CVPR22]	`ResNet-18`	Youtube-VOS(- , 5.58 hours)	72.1	69.7	81.4	74.5	85.9
OURS	`ResNet-18`	Youtube-VOS(- , 5.58 hours)	**74.5**	**71.6**	**82.9**	**77.4**	**86.9**
	`ResNet-50`		**75.6**	**73.3**	**83.6**	**77.8**	**87.3**
OSVOS[12][CVPR17]	`VGG-16`	I+D(1.28M, 10k)	60.3	56.6	63.8	63.9	73.8
STM[10][ICCV19]	`ResNet-50`	I+D+Youtube-VOS(1.28M, 164k)	81.8	79.2	88.7	84.3	91.8

- I: ImageNet [75]; C: COCO [76]; D: DAVIS$_{17}$ [42].

Table 1. **Quantitative segmentation results** (§4.1) on DAVIS$_{17}$ [42] `val`. For dataset size, we report (#*raw* images, length of *raw* videos) for self-supervised methods and (#image-level annotations, #pixel-level annotations) for supervised methods.

4. Experiments

Dataset. We evaluate our approach on two VOS datasets, *i.e.*, DAVIS$_{17}$ [42] and YouTube-VOS [52]. They have 30 and 474 videos in `val` sets, respectively. The videos are companied with pixel-wise annotations and cover various challenges like occlusion, complex background, and motion blur.

Evaluation Metric. Following the official evaluation protocols [42, 52], we adopt region similarity ($\mathcal{J}_m$), contour accuracy ($\mathcal{F}_m$) and their average ($\mathcal{J}\&\mathcal{F}_m$). For DAVIS$_{17}$, we additionally report the recall values ($\mathcal{J}_r$ and $\mathcal{F}_r$), at IoU threshold 0.5. For YouTube-VOS, scores are obtained by submitting the results to the official evaluation server and separately computed for *seen* and *unseen* categories.

4.1. Comparison with State-of-the-Art

Performance on DAVIS$_{17}$. Table 1 gives comparison results against 15 recent self-supervised VOS methods on DAVIS$_{17}$ `val`. We also include two famous supervised alternatives [10, 12] for reference. As seen, using a relatively small amount of training data (*i.e.*, 5.58 hours of raw videos in YouTube-VOS `train`) and weak backbone architecture – `ResNet-18`, our approach outperforms all competitors across multiple evaluation metrics. When adopting `ResNet-50`, our approach yields far better performance, up to **75.6**% $\mathcal{J}\&\mathcal{F}_m$. In particular, compared with `ResNet-18` based top-leading models, *i.e.*, LIIR [31], SCR [40], DUL [38], and CLTC [37],

Method	Backbone	$\mathcal{J}\&\mathcal{F}_m\uparrow$	$\mathcal{J}_m\uparrow$	$\mathcal{J}_r\uparrow$	$\mathcal{F}_m\uparrow$	$\mathcal{F}_r\uparrow$
MAST[28] [CVPR20]	ResNet-18	54.3	50.7	58.9	57.8	64.5
CRW[34] [NeurIPS20]	ResNet-18	55.9	52.3	-	59.6	-
DUL[38] [NeurIPS21]	ResNet-18	57.0	53.5	60.4	60.5	67.6
SCR[40] [CVPR22]	ResNet-18	59.9	55.9	-	64.0	-
LIIR[31] [CVPR22]	ResNet-18	57.5	55.2	63.1	59.8	68.6
OURS	ResNet-18	**61.3**	**59.4**	**66.5**	**63.1**	**73.7**
	ResNet-50	**62.4**	**60.6**	**66.9**	**64.2**	**74.3**
RGMP[17] [CVPR18]	ResNet-50	52.9	51.3	-	54.4	-
STM[10] [ICCV19]	ResNet-50	72.2	69.3	-	75.2	-

Table 2. **Quantitative results** (§4.1) on DAVIS$_{17}$ [42] test-dev.

our approach earns **2.4**%, **4.0**%, **5.2**%, and **4.2**% $\mathcal{J}\&\mathcal{F}_m$ gains, respectively. Note that, CLTC adopts different network architectures and model weights for different datasets. Apart from this, VFS and JSTG make use of much more training data than ours (800 *vs*. 5.58 hours of videos). As for DINO, a recent state-of-the-art, contrastive image representation learning based method, our approach still outperforms it by **3.1**% and **4.2**% $\mathcal{J}\&\mathcal{F}_m$ based on ResNet-18 and ResNet-50, respectively. This is parti- cularly impressive, considering our backbone is *desperately inferior* to DINO (*i.e*., ResNet-18/-50 *vs*. ViT-B) and the training data used by these two methods are completely not comparable in both quality and quantity (*i.e*., 3.5K videos *vs*. 1.28M images). When using the same ResNet-50 back- bone, the performance gap is huge, *e.g*., **19.4%** in $\mathcal{J}\&\mathcal{F}_m$. Table 2 reports our performance on DAVIS$_{17}$ test-dev. We can clearly observe that, our approach, again, suppresses all the recent alternatives by a solid margin.

Performance on YouTube-VOS. We further conduct experiments on YouTube-VOS val. As shown in Table 3, our approach, again, achieves remarkable performance, evidencing its efficacy and generalization ability across different VOS datasets. Specifically, when opting for ResNet-18 backbone network architecture, our approach obtains **1.7**% absolute $\mathcal{J}\&\mathcal{F}_m$ improvement, over the current top leading method — DUL. Moreover, with a stronger backbone — ResNet-50, our approach further improves the $\mathcal{J}\&\mathcal{F}_m$ score to **72.4**%, setting a new state-of-the-art.

Visual Comparison Results. Fig. 3 depicts the visual comparison results of our approach and two competitors, MAST and DUL, on two challenging videos from DAVIS$_{17}$ val and YouTube-VOS val, respectively. We can find CRW and LIIR, as classic, correspondence-based methods, suffer from drifting errors during mask propagation; small prediction errors on past frames are hard to be corrected in later frames and further lead to worse results after processing more frames. This is due to their matching-based propagation strategy. In contrast, our approach generates more reasonable segments that better align object boundaries, and performs robust to small outlier predictions, hence reducing error accumulation over time. These results verify the efficacy of our model and support our insight that encoding mask information is crucial for self-supervised VOS. Further detailed quantitative analyses can be found in §4.2.

Method	Backbone	$\mathcal{J}\&\mathcal{F}_m\uparrow$	Seen		Unseen	
			$\mathcal{J}_m\uparrow$	$\mathcal{F}_m\uparrow$	$\mathcal{J}_m\uparrow$	$\mathcal{F}_m\uparrow$
Colorization[26] [ECCV18]	ResNet-18	38.9	43.1	38.6	36.6	37.4
CorrFlow[27] [IMVC19]	ResNet-18	46.6	50.6	46.6	43.8	45.6
MAST[28] [CVPR20]	ResNet-18	64.2	63.9	64.9	60.3	67.7
CRW[34] [NeurIPS20]	ResNet-18	68.7	67.4	69.1	65.1	73.2
CLTC[37] [CVPR21]	ResNet-18	67.3	66.2	67.9	63.2	71.7
DUL[38] [NeurIPS21]	ResNet-18	69.9	69.6	71.3	65.0	73.5
LIIR[31] [CVPR22]	ResNet-18	69.3	67.9	69.7	65.7	73.8
OURS	ResNet-18	**71.6**	**71.0**	**74.2**	**66.0**	**75.3**
	ResNet-50	**72.4**	**71.7**	**74.6**	**67.0**	**76.2**
OSVOS[12] [CVPR17]	VGG-16	58.8	59.8	60.5	54.2	60.7
STM[10] [ICCV19]	ResNet-50	79.4	79.7	84.2	73.5	80.9

Table 3. **Quantitative results** (§4.1) on YouTube-VOS [52] val.

4.2. Diagnostic Experiments

To thoroughly examine our core hypotheses and model designs, we conduct a series of ablative studies on DAVIS$_{17}$ val. The reported baselines are built upon ResNet-18 and trained by the default setting, unless otherwise specified.

Training Objective. Our model is jointly trained for mask-embedded segmentation $\mathcal{L}_{\text{Seg}}$ (*cf*. Eq. 16) and correspondence matching $\mathcal{L}_{\text{Corr}}$ ($=\mathcal{L}_{\text{Short}}+\mathcal{L}_{\text{Long}}$). Table 4a analyzes the influence of different training objectives. We can find that, using $\mathcal{L}_{\text{Short}}$ or $\mathcal{L}_{\text{Long}}$ individually only yields $\mathcal{J}\&\mathcal{F}_m$ scores of 57.4% and 67.2%, respectively. Their combination uplifts the performance to 68.8%, confirming their complementarity. However, the baseline is still weaker in comparison with current top-leading correspondence-based methods, *e.g*., LIIR [31] with 72.1%. Moreover, when us-

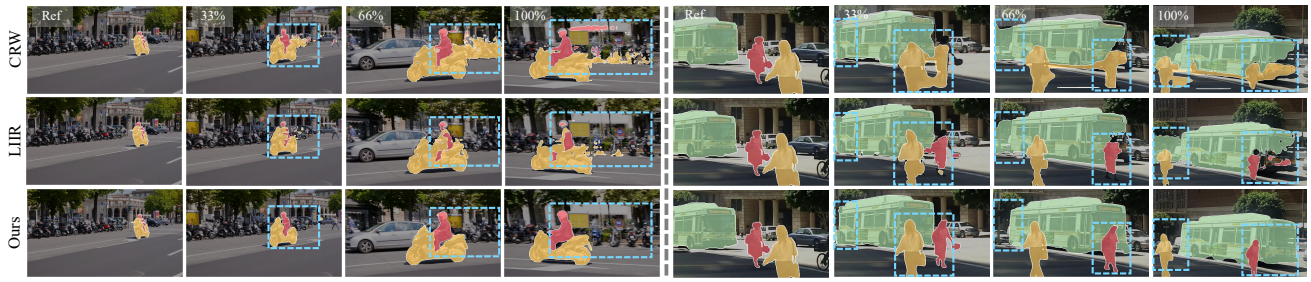

Figure 3. **Visual comparison results** (§4.1) on two videos from DAVIS$_{17}$ [42] val (left) and Youtube-VOS [52] val (right), respectively. CRW [34] and LIIR [31] suffer from error accumulation during mask tracking, due to the simple matching-based mask copy-paste strategy. However, our approach performs robust over time and yields more accurate segmentation results, by learning to embed target masks.

Loss	$\mathcal{J}\&\mathcal{F}_m\uparrow$	$\mathcal{J}_m\uparrow$	$\mathcal{F}_m\uparrow$
$\mathcal{L}_{\text{Short}}$	57.4	55.8	58.9
$\mathcal{L}_{\text{Long}}$	67.2	64.9	69.5
$\mathcal{L}_{\text{Short}}$ + $\mathcal{L}_{\text{Long}}$	68.8	66.7	70.9
$\mathcal{L}_{\text{Seg}}$	62.3	60.5	64.0
$\mathcal{L}_{\text{Seg}}$+$\mathcal{L}_{\text{Short}}$+$\mathcal{L}_{\text{Long}}$	**74.5**	**71.6**	**77.4**

(a) loss design

#Ref. Frame	$\mathcal{J}\&\mathcal{F}_m\uparrow$	$\mathcal{J}_m\uparrow$	$\mathcal{F}_m\uparrow$
First	68.8	65.7	71.9
First + Last 1:15	73.2	70.4	76.0
First + Last 1:20	**74.5**	**71.6**	**77.4**
First + Last 1:25	73.5	70.9	76.1
First + Last 1:30	72.8	70.2	75.3

(b) number of reference frames

#Centroid	$\mathcal{J}\&\mathcal{F}_m\uparrow$	$\mathcal{J}_m\uparrow$	$\mathcal{F}_m\uparrow$
$M = 2$	67.5	65.2	69.8
$M = 3$	71.6	69.0	74.2
$M = 5$	**74.5**	**71.6**	**77.4**
$M = 8$	72.5	69.6	75.4
$M = 10$	70.1	67.3	72.9

(c) number of cluster centers

Mask update	$\mathcal{J}\&\mathcal{F}_m\uparrow$	$\mathcal{J}_m\uparrow$	$\mathcal{F}_m\uparrow$
No update	71.1	68.3	73.9
Per 20 epoch	72.8	69.9	75.7
Per 15 epoch	73.9	70.8	77.0
Per 10 epoch	**74.5**	**71.6**	**77.4**
Per 5 epoch	72.5	69.5	75.5
Every epoch	69.7	66.7	72.6

(d) pseudo mask update

Round	$\mathcal{J}\&\mathcal{F}_m\uparrow$	$\mathcal{J}_m\uparrow$	$\mathcal{F}_m\uparrow$	FPS
0	69.7	67.3	72.1	1.86
1	72.6	69.8	75.4	1.84 (-1.1%)
2	73.9	71.1	76.7	1.80 (-3.2%)
3	**74.5**	**71.6**	**77.4**	1.77 (-4.8%)
4	74.3	71.2	**77.3**	1.73 (-7.0%)
5	74.0	71.0	77.0	1.69 (-9.2%)

(e) recurrent refinement

Strategy	Loss	$\mathcal{J}\&\mathcal{F}_m\uparrow$	$\mathcal{J}_m\uparrow$	$\mathcal{F}_m\uparrow$	FPS
photometric	MAST [28]	65.5	63.3	67.6	1.13
reconstruction	MAST [28]+$\mathcal{L}_{\text{Seg}}$	**69.0** (+3.5)	**66.4**	**71.6**	1.01
cycle-consistency	CRW [34]	67.6	64.6	70.6	1.86
tracking	CRW [34]+$\mathcal{L}_{\text{Seg}}$	**71.8** (+4.2)	**68.3**	**75.3**	1.77
contrastive	$\mathcal{L}_{\text{Corr}}$ (ours)	68.8	66.7	70.9	1.86
matching	$\mathcal{L}_{\text{Corr}}$+$\mathcal{L}_{\text{Seg}}$	**74.5** (+5.7)	**71.6**	**77.4**	1.77

(f) correspondence learning schema

Table 4. A set of ablative studies on DAVIS$_{17}$ [42] `val` (§4.2). The adopted settings are marked in red.

ing $\mathcal{L}_{\text{Seg}}$ solely, the model only achieves 62.3%. This is because, without the regularization of the correspondence learning term, k-means suffers from random initialization of the representation and easily return trivial solutions, *e.g.*, fragile or massive clusters. When considering all the training goals together, performance boosts can be clearly observed, *e.g.*, **74.5%** in $\mathcal{J}\&\mathcal{F}_m$. Under such a scheme, unsupervised correspondence learning makes the features informative for meaningful clustering; then the produced high-quality pseudo masks allow the model to learn to make a better use of the object mask to guide segmentation.

Reference Frame. As usual [9, 10, 28], we leverage the first frame and several previous segmented frames as well as their corresponding masks, to support the segmentation of the current frame. Table 4b reports the related experiments.

k-means Clustering. Next we probe the impact of the number of cluster centers, *i.e.*, M, in Table 4c. The best performance is obtained at $M=5$, roughly equal to the obvious objects number, *i.e.*, $3 \sim 4$ on average in each training video.

Pseudo Mask Update. During training, our approach alternates between clustering based pseudo mask generation and mask guided segmentation learning. In Table 4d, we study such training strategy. 'No update' means that, after the initial correspondence learning stage (first 300 training epochs; see §3.3), we create pseudo masks and use them throughout the whole joint correspondence and segmentation learning stage (last 100 epochs). This baseline achieves 71.1% $\mathcal{J}\&\mathcal{F}_m$. If we improve the frequency of pseudo mask update from once to twice every 20 epochs, the score is improved to **74.5**%. But further more frequently re-estimating the pseudo masks leads to inferior performance. We speculate that it is because, when learning with the noisy pseudo masks, it needs more epochs to optimize the network parameters, while updating the pseudo masks too frequently will easily suffer from the impact of sub-optimal features.

Recurrent Refinement. We feed our predicted masks to the segmentation decoder $\mathcal{D}$ for iterative refinement. Table 4e reports the related results. *Round 0* means we follow Eq. 7 to leverage $\boldsymbol{V}_q$ for mask decoding. In *Round 1*, the model follows Eq. 9 to warp and refine the coarse prediction $\bar{Y}_q$ and from *Round 2* onwards, we replace $\bar{Y}_q$ with the output $\hat{Y}_q$ from the prior round. As seen, after two rounds of refinement, $\mathcal{J}\&\mathcal{F}_m$ score is improved from 69.7% to **74.5**%, with only negligible delay in inference speed (*i.e.*, -4.8%).

Versatility. As our self-supervised mask embedding learning (*cf.* §3.1) is a general framework, it is interesting to test its efficacy with different correspondence learning regimes (*cf.* §2). In Table 4f, we apply our mask embedding learning method to MAST [28] (reconstruction based), CRW [34] (cycle-consistency based), and our correspondence learning strategy $\mathcal{L}_{\text{Corr}}$ (*cf.* §3.2; contrastive matching based). Impressively, notable performance gains are achieved over different baselines, *e.g.*, **3.5%** on MAST, **4.2%** on CRW, and **5.7%** on our $\mathcal{L}_{\text{Corr}}$, in terms of $\mathcal{J}\&\mathcal{F}_m$. The last column of Table 4f gives inference speed, showing the additional computational budget brought by mask embedding is negligible.

5. Conclusions

Current solutions for self-supervised VOS are commonly built upon unsupervised correspondence matching, detached from the mask-guided, sequential segmentation nature of the task. In contrast, we devised a new framework that investigates both mask embedding and correspondence learning for mask propagation, in an annotation-free manner. Through space-time clustering, coherent video partitions are automatically generated for teaching the model to directly learn mask embedding and tracking. Meanwhile, self-supervised correspondence learning is naturally incorporated as extra regularization. In this way, our approach successfully bridges the gap between fully- and self-supervised VOS models in both performance and network architecture design.

Acknowledgements. This work was supported by Beijing Natural Science Foundation under Grant L191004 and the Fundamental Research Funds for the Central Universities (No. 226-2022-00051).

References

[1] Tianfei Zhou, Fatih Porikli, David J Crandall, Luc Van Gool, and Wenguan Wang. A survey on deep learning technique for video segmentation. *IEEE TPAMI*, 2022. 1, 2

[2] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. Video propagation networks. In *CVPR*, 2017. 1

[3] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017.

[4] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018. 2

[5] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 2

[6] Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. Video object segmentation with episodic graph memory networks. In *ECCV*, 2020. 2

[7] Jiaxu Miao, Yunchao Wei, and Yi Yang. Memory aggregation networks for efficient interactive video object segmentation. In *CVPR*, 2020.

[8] Tim Meinhardt and Laura Leal-Taixé. Make one-shot video object segmentation efficient again. In *NeurIPS*, 2020.

[9] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *NeurIPS*, 2021. 2, 4, 6, 8

[10] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 2, 4, 6, 7, 8

[11] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *IEEE TPAMI*, 44(9):4701–4712, 2021. 2

[12] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 2, 6, 7

[13] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *TPAMI*, 41(6):1515–1530, 2018. 2

[14] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *NeurIPS*, 2021.

[15] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *ECCV*, 2020. 2, 3

[16] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *CVPR*, 2020. 2

[17] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018. 2, 7

[18] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019. 2

[19] Ruizheng Wu, Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. Memory selection network for video propagation. In *ECCV*, 2020.

[20] Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *CVPR*, 2021. 2

[21] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *CVPR*, 2021.

[22] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. In *NeurIPS*, 2020. 2

[23] Zhengdong Hu, Yifan Sun, and Yi Yang. Switch to generalize: Domain-switch learning for cross-domain few-shot classification. In *ICLR*, 2022.

[24] Wenguan Wang, Jianbing Shen, Fatih Porikli, and Ruigang Yang. Semi-supervised video object segmentation with super-trajectories. *IEEE TPAMI*, 41(4):985–998, 2018.

[25] Wenguan Wang, Jianbing Shen, Xiankai Lu, Steven CH Hoi, and Haibin Ling. Paying attention to video object pattern understanding. *IEEE TPAMI*, 43(7):2413–2428, 2020. 1

[26] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018. 1, 3, 6, 7

[27] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. In *BMVC*, 2019. 6, 7

[28] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *CVPR*, 2020. 1, 3, 6, 7, 8

[29] Youngeun Kim, Seokeon Choi, Hankyeol Lee, Taekyung Kim, and Changick Kim. Rpm-net: Robust pixel-level matching networks for self-supervised video object segmentation. In *WACV*, 2020.

[30] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *ICCV*, 2021.

[31] Liulei Li, Tianfei Zhou, Wenguan Wang, Lu Yang, Jianwu Li, and Yi Yang. Locality-aware inter-and intra-video reconstruction for self-supervised correspondence learning. In *CVPR*, 2022. 1, 6, 7

[32] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 1, 3, 6

[33] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, 2019.

[34] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 1, 3, 6, 7, 8

[35] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *NeurIPS*, 2019. 1, 3, 6

[36] Zhangxing Bian, Allan Jabri, Alexei A Efros, and Andrew Owens. Learning pixel trajectories with multiscale contrastive random walks. *arXiv preprint arXiv:2201.08379*, 2022. 1

[37] Sangryul Jeon, Dongbo Min, Seungryong Kim, and

Kwanghoon Sohn. Mining better samples for contrastive learning of temporal correspondence. In *CVPR*, 2021. 1, 3, 5, 6, 7

[38] Nikita Araslanov, Simone Schaub-Meyer, and Stefan Roth. Dense unsupervised learning for video segmentation. In *NeurIPS*, 2021. 3, 5, 6, 7

[39] Jiarui Xu and Xiaolong Wang. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *ICCV*, 2021. 1, 3, 5, 6

[40] Jeany Son. Contrastive learning for space-time correspondence via self-cycle consistency. In *CVPR*, 2022. 1, 6, 7

[41] Ning Wang, Wengang Zhou, and Houqiang Li. Contrastive transformation for self-supervised correspondence learning. In *AAAI*, 2021. 1, 3, 6

[42] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 1, 2, 6, 7, 8

[43] Vijay Badrinarayanan, Fabio Galasso, and Roberto Cipolla. Label propagation in video sequences. In *CVPR*, 2010. 2

[44] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014.

[45] S Avinash Ramakanth and R Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *CVPR*, 2014.

[46] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Selective video object cutout. *IEEE TIP*, 26(12):5645–5655, 2017.

[47] Wenguan Wang, Jianbing Shen, Jianwen Xie, and Fatih Porikli. Super-trajectory for video segmentation. In *ICCV*, 2017. 2

[48] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 2

[49] Qingnan Fan, Fan Zhong, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Jumpcut: non-successive mask transfer and interpolation for video cutout. *TOG*, 34(6):195–1, 2015.

[50] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015. 2

[51] Xiushan Nie Dongfang Liu Yilong Yin Wenguan Wang Zheyun Qin, Xiankai Lu. Coarse-to-fine video instance segmentation with factorized conditional appearance flows. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1–17, 2023. 2

[52] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 2, 4, 6, 7

[53] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin. A transductive approach for video object segmentation. In *CVPR*, 2020. 2

[54] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, pages 629–645, 2020. 2, 4

[55] Li Hu, Peng Zhang, Bang Zhang, Pan Pan, Yinghui Xu, and Rong Jin. Learning position and target consistency for memory-based video object segmentation. In *CVPR*, 2021. 2

[56] Zhengdong Hu, Yifan Sun, and Yi Yang. Suppressing the heterogeneity: A strong feature extractor for few-shot segmentation. In *ICLR*, 2023. 2

[57] Yunyao Mao, Ning Wang, Wengang Zhou, and Houqiang Li. Joint inductive and transductive learning for video object segmentation. In *ICCV*, 2021. 2

[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[59] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J Crandall, and Steven CH Hoi. Learning video object segmentation from unlabeled videos. In *CVPR*, 2020. 3, 5, 6

[60] Zixu Zhao, Yueming Jin, and Pheng-Ann Heng. Modelling neighbor relation in joint space-time graph for video correspondence learning. In *ICCV*, 2021. 6

[61] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *ECCV*, 2020. 3

[62] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[63] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 3

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

[65] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 4

[66] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 5

[67] Jarmo Hurri and Aapo Hyvärinen. Simple-cell-like receptive fields maximize temporal coherence in natural video. *Neural Computation*, 15(3):663–691, 2003. 5

[68] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *CVPR*, 2018. 5

[69] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *ICCV*, 2017.

[70] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NeurIPS*, 2017.

[71] Pedro O O Pinheiro, Amjad Almahairi, Ryan Benmalek, Florian Golemo, and Aaron C Courville. Unsupervised learning of dense visual representations. *NeurIPS*, 2020. 5

[72] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *ICML*, 2009. 5

[73] Yichao Yan, Ning Zhuang, Jian Zhang, Minghao Xu, Qiang Zhang, Zhang Zheng, Shuo Cheng, Qi Tian, Xiaokang Yang, Wenjun Zhang, et al. Fine-grained video captioning via graph-based multi-granularity interaction learning. *IEEE TPAMI*, 2019. 5

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

ICCV, 2021. 6

[75] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

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