Learning segmentation from point trajectories

Laurynas Karazija, Iro Laina, Christian Rupprecht, Andrea Vedaldi
Visual Geometry Group
University of Oxford
Oxford, UK
{laurynas,iro,chrisr,vedaldi}@robots.ox.ac.uk

Abstract

We consider the problem of segmenting objects in videos based on their motion and no other forms of supervision. Prior work has often approached this problem by using the principle of common fate, namely the fact that the motion of points that belong to the same object is strongly correlated. However, most authors have only considered instantaneous motion from optical flow. In this work, we present a way to train a segmentation network using long-term point trajectories as a supervisory signal to complement optical flow. The key difficulty is that long-term motion, unlike instantaneous motion, is difficult to model – any parametric approximation is unlikely to capture complex motion patterns over long periods of time. We instead draw inspiration from subspace clustering approaches, proposing a loss function that seeks to group the trajectories into low-rank matrices where the motion of object points can be approximately explained as a linear combination of other point tracks. Our method outperforms the prior art on motion-based segmentation, which shows the utility of long-term motion and the effectiveness of our formulation.

1 Introduction

Segmentation, the task of delineating and isolating distinct objects, is a fundamental problem in computer vision. Much of the current approaches are supervised, relying on expensive manual annotations. Attempts to approach this task without supervision have largely relied on manual heuristics or exploited the rich semantics of self-supervised feature extractors. Video data, however, offers an additional option as it contains *motion*, which can be exploited for an additional inductive bias. Such approaches are rooted in the principle of common fate from Gestalt psychology [66], which posits that elements that move together are more likely to belong together.

Motion information is usually captured by optical flow. Flow is attractive as it arises from low-level visual properties and can provide a signal before scenes are parsed and objects are discovered. Furthermore, optical flow estimators, such as RAFT [60] or FlowFormer [24], can be trained purely on synthetic artificial data, transferring to real-world scenes with remarkable accuracy and without manual annotation. This has led many to consider optical flow as a critical modality to discover and learn objects from video data by learning to attribute and explain the motions of objects.

Optical flow, however, only describes the instantaneous motion of the scene, which can create blindspots: not all objects are necessarily in motion at all times. Similarly, groups of objects might coincidentally move together. Recent advances in point tracking [14, 15, 22, 28] offer an alternative form of motion information. Point trackers "lock on" to a set of query points and describe their position and visibility over the course of the whole video. This provides long-term motion information. Like optical flow estimators, point trackers are trained on synthetic data. However, unlike optical flow, point trajectories describe only a sparse set of points.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3576

In this paper, we ask whether the long-term motion information obtained in point trajectories is beneficial. To that end, we explore how to supervise image segmentation networks using motion information with point trajectories. At a glance, this presents several problems. Firstly, point trajectories are time-varying 2D point clouds, and combining them with image-based networks is not straightforward. Furthermore, the evolution of long-term object motion is too complex, even in the simplest cases. Our main insight is that the motion of points belonging to the same object should be well correlated. We thus propose a loss function that encodes this intuition by seeking to explain groups of points as combinations of other points in the group. With our method, a segmentation network predicts objects in the scene, inducing a grouping of trajectories that are currently visible. The loss function then assesses how well such a grouping explains the long-term motion. While point trajectories describe motion over a longer time, they are limited by the number of points tracked, which is often much less than the number of pixels. We thus propose to train using both trajectory-based loss and optical flow-based loss and show that spatially sparse but longer-time motion information synergises with spatially dense optical flow.

Discovering objects using point trajectories has a long history in computer vision. Our approach is inspired by ideas of subspace clustering, which assume that data comes from distinct subspaces and seek to reconstruct membership information of data points. This has previously been applied to the problem of motion segmentation [17, 38]. These approaches, however, are sensitive to noise and either rely on specialised optimisation procedures to recover a graph of trajectory relationships [32, 48] or use manual instead [32, 48]. Normalised cuts or spectral clustering are then used to group the trajectories. However, the need for an affinity matrix limits the number of trajectories that can be used due to quadratic memory requirements. Furthermore, "densification" is still required to extend trajectory clusters to the whole image. By construction, these approaches can process only a single sequence at a time. Our proposal instead trains an image segmentation network directly end-to-end using a dataset of videos while supporting a large number of trajectories.

In summary, our work makes the following contributions. (1) We propose a loss function that enables training any image segmentation architecture using point trajectories as a source of supervision. (2) We investigate our proposed loss in a principled way in a simulated setting, showing the feasibility of our approach. (3) We apply such a loss in a per-sequence optimisation, outperforming other subspace clustering baselines. (4) We use our loss to train a single network on a dataset of videos for the task of video object segmentation, demonstrating strong results. (5) We show how our proposed loss formulation obtains better performance than alternatives.

2 Related work

Unsupervised video object segmentation. Video object segmentation (VOS) aims to label pixels of objects in a video. Current VOS benchmarks [35, 51, 53] usually define the problem as binary foreground-background separation or salient object segmentation. The task is usually approached in two ways: semi-supervised and unsupervised VOS. Semi-supervised methods require initial frame annotations and aim to *propagate* them to the rest of the video [7]. Unsupervised VOS aims to discover object(s) of interest without the initial targets [18, 25, 36, 42, 52, 61]. This however does not differentiate methods based on data used to *train* them. Most of the traditional research in semi- or unsupervised VOS relies on annotations during training. Our approach, in contrast, does not rely on any manual annotations to learn. Some authors explore related unsupervised video instance segmentation [65] task without any annotations, object-centric learning appraoches [2, 58, 73], some of which make use of flow [29] and depth [57].

Motion segmentation. A closely related task to video object segmentation is motion segmentation, which aims to extract the main moving objects in a video. The practical difference between these two tasks is more difficult to delineate as the same benchmark datasets are often used. Early works modeled the scenes as layers [8, 27], which later works accomplish using a slot-attention mechanism [13, 34, 69]. Flow mixture models accounted for multiple motion patterns [26, 62], and corrections were introduced for rotating cameras [3, 4]. Later works [10, 46, 47] considered parametric flow models fit to explain the scene. AMD [40] employs a single model with separate appearance and motion 'pathways'. Other works train flow-only models by generating synthetic data, which generalise well to real videos [33, 67]. An alternative line of work adopts a more generative approach, training an inpainter networks to predict optical flow [70, 71]. Several authors [37, 59]

adopt a multi-stage self-labelling [65] approach for motion segmentation: initial masks are estimated using an optical flow-based approach, followed by DINO-based refinement and CRF post-processing to generate pseudo-labels and train a final segmentation network.

Trajectory-based motion segmentation. Trajectory-based motion segmentation has also been explored. Older works consider data of multiple trajectories and employ non-negative matrix factorization and related decomposition methods [9, 11, 17, 19, 55, 68]. This line of work primarily operates by defining affinity between pairwise trajectories in a single video setting. In [6, 30, 31, 49, 50], heuristic graphs are constructed between trajectories, considering increasingly complex motion models, and employing specialised solvers to solve the optimisation problem. However, due to the specialised optimisation procedures and tight coupling with trajectory estimation methods, this line of work has received less attention than deep methods that exploit optical flow similarly to RGB frames.

Subspace clustering. A specific kind of trajectory-based technique is subspace clustering approaches, which rely on the *self-expressive* property of the data. They can largely be summarised [21] as solving a constrained optimisation problem $\arg\min_C ||DC - D||_F^2 + \lambda\theta(C)$ for some dataset $D \in \mathbb{R}^{d\times n}$ of n points in d dimensions. C is a matrix of coefficients, which expresses the data and can be represented as a linear combination of other points. Given a solution for the coefficient matrix, it is transformed into an affinity matrix for spectral clustering. The approaches mainly differ in the second term of the objective and specialized methods to solve the optimisation problem. SSC [17] define $\theta(C)$ as l_1 norm. LLR [38] use nuclear norm instead, while LSR [41] uses instead l_2 regularisation. [41, 43, 64] combines l_1, l_2, and nuclear norms. Under some strong assumptions [21], these approaches enjoy some theoretical guarantees. However, they are difficult to scale in practice as the number of points n grows, as C is $n \times n$. Additionally, the secondary step of spectral clustering is also limiting and difficult to tune. Instead, we take inspiration from these approaches and propose a way to supervise the network directly using the self-expressive property of point trajectories.

3 Method

Our goal is to solve the video segmentation task in an unsupervised manner: given a video, we want to segment out the objects that are moving independently within it. A video is a sequence of frames $\mathcal{I}_t \in \mathbb{R}^{HW\times 3}$, each of which is an RGB image defined on the lattice $\Omega = \text{vec}(\{1,\dots,H\} \times \{1,\dots,W\}) \in \mathbb{R}^{HW\times 1}$. To segment the objects, we self-supervise a neural network Φ that takes as input each frame $\mathcal{I}_t$ in turn, and outputs a corresponding segmentation mask $\Phi(\mathcal{I}_t) = M_t \in [0,1]^{HW\times K}$ where K is the number of possible segments we expect to observe in the video. Segmentation matrix entries softly assign each pixel to one of K possible segments.

The challenge is how to supervise the network Φ without labels, utilising only the video itself as training material. The key inductive principle that we propose to use is that physical points that belong to the same object tend to have highly correlated motion, often called *principle of common fate*. When these points are projected to pixels, they result in corresponding highly correlated apparent motions, which we can measure using techniques like optical flow and point tracking. Therefore, we propose to supervise the network Φ from an analysis of apparent motion extracted automatically from the video using off-the-shelf components.

Motion can be measured at two temporal scales. Optical flow extracts instantaneous motion, measuring the 2D velocity of the 3D points found at each pixel in each video frame. Point tracking extracts long-term motion, estimating the 2D location of a certain number of 3D points throughout the video's duration. These two sources of information are complementary. Optical flow is dense, easy to extract, and easy to model to discover correlations within it; however, by considering different times in isolation, it ignores most of the correlations that exist in the data. Tracks are sparse, more difficult to extract and harder to model, but potentially contain information ignored by optical flow.

Prior works such as [10] have studied how to model optical flow for segmentation. Here, motivated by a new generation of high-quality point trackers [14, 15, 22, 28], we aim at developing the machinery necessary to use track information as well. From this analysis, we construct losses which assess the quality of the predicted mask M_t given the video itself. Next, we introduce two such losses, one for optical flow from prior work, and a new one based on point tracking.

 https://doi.org/10.52202/079017-3576

3.1 Learning from optical flow

First, we describe the case of optical flow. Because optical flow is instantaneous, we can fix our attention on a specific frame $\mathcal{I}$ and corresponding mask M, dropping for now the time index t. The *optical flow* $F \in \mathbb{R}^{HW \times 2}$ for this image associates a 2-dimensional flow vector to each of the $H \times W$ pixels. Each flow vector can be understood as the velocity of the pixel.

Let $M_k \in \mathbb{R}^{HW \times 1}$ be the binary matrix for segment k, obtained by extracting the k-th column of M. Let $F_k = M_k \odot F$ denote the Hadamard (element-wise) product between the mask and flow vectors, broadcasting the mask along the rows.

Assuming that the object is rigid, the optical flow can be approximated as a linear parametric model of 2D coordinate embeddings (see [1] for an overview). Following [10], we consider a six-dimensional quadratic embedding kernel $\mathrm{emb}([x,\ y]) = [x,\ x^2,\ y,\ y^2,\ xy,\ 1] \in \mathbb{R}^{1\times 6}$ for pixel coordinates $[x,\ y] \in \Omega$ and associate to each region k a corresponding set of 12 parameters $\theta_k \in \mathbb{R}^{6\times 2}$. Optical flow vectors within a region should be expressible as a linear combination of these six basis functions.

We then consider all pixels embeddings stacked in a single matrix $E_k = M_k \odot \mathrm{emb}(\Omega)$ where the product with the soft mask ensures that the embeddings are "active" only if the corresponding pixels are. The optical flow vectors in the region are then approximated as

$$F_k \approx \hat{F}_k = E_k \hat{\theta}_k \quad \text{where} \quad \hat{\theta}_k = (E_k^\top E_k)^{-1} E_k^\top F_k, \tag{1}$$

where $\hat{\theta}_k$ is obtained via least square. We can use the residual of this approximation as a measure of how well the mask M_k fits the data:

$$\mathcal{L}_f(M|F) = \sum_k \|F_k - \hat{F}_k\|_F^2 = \sum_k \|F_k - E_k\hat{\theta}_k\|_F^2 \,. \tag{2}$$

Intuitively, this considers the correlation of pixel motion in the *spatial* sense: how pixel coordinates determine its motion based on motion parameters θ_k.

3.2 Learning from trajectories

Having covered optical flow, we move now to developing an analogous loss for tracking. We write $P \in \mathbb{R}^{2T\times N}$ for the track matrix, with one trajectory per column. With slight abuse of notation, we write $(P)_t \in \mathbb{R}^{2\times N}$ for indexing rows corresponding to point locations at some time t. To connect pixel-wise masks and sparse points, we use a sampling operation $\pi(\cdot)$, writing $\pi(M_k, (P)_t) = \hat{M}_k \in [0,1]^{N\times 1}$ for mask values at point locations at an appropriate time. Furthermore, we denote by $P_k = P \odot \hat{M}_k$ the masked version of the trajectory matrix, selecting the columns/trajectories that belong to object k with obvious broadcasting of the mask values.

Unlike optical flow, trajectories are too complex to be modelled using a small set of *fixed* basis functions. Instead, we posit that the set of trajectories should be low-rank — all trajectories belonging to the same object should be explained well by a linear combination of some small number of trajectories. We illustrate this intuition in Fig. 1 using a 2D example.

This assumption results in a factorization of P_k using singular value decomposition (SVD) as $P_k = U_k \Sigma_k V_k^\top$, where $(U_k, \Sigma_k, V_k) = \mathrm{SVD}(P_k)$. As P_k should be low-rank, we can thus form an approximation using truncated SVD, by considering only first r components. We write $\lfloor U_k \rfloor_r$ to denote such truncation. With this, we obtain the loss

$$\mathcal{L}_{\mathrm{rec@}r}(M|P) = \sum_k \left\| P_k - \lfloor U_k \rfloor_r \lfloor \Sigma_k \rfloor_r \lfloor V_k \rfloor_r^\top \right\|_F^2 . \tag{3}$$

Since truncated SVD offers optimal decomposition for the error above, lowering this loss amounts to making P_k as close as possible to rank r, i.e., by grouping trajectories into P_k that do not increase its rank, and should come from rigid objects.

As we show in Section 5.3, we found an alternative formulation of this idea works better. Note the rank r matrix has the r-th and all later singular values as 0. We can optimise singular values higher than r-th to be close to 0 (ignoring U_k and V_k). Thus, for trajectories, we formulate a loss simply as:

$$\mathcal{L}_t(M|P) = \sum_k \sum_{i=r}^{\min(2T,N)} \sigma_i(P_k), \tag{4}$$

where $\sigma_i(P_k)$ is the i-th singular value of P_k. We assume $r \ll \min(2T, N)$.

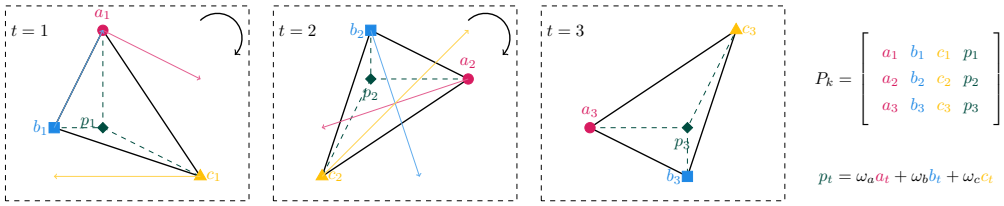


Figure 1: **Illustrative 2D example for the low-rank nature of P_k.** A triangle undergoes rigid rotation over three frames. As the rate of rotation is not constant, the flow vectors and point positions are difficult to model. However, the point p is part of the triangle and can be expressed as a combination of the three vertices at an appropriate time. Thus, the last column of P_k is linearly dependent, and P_k is rank deficient. *Any points in the triangle could be included in P_k without increasing its rank.*

Meaning of decomposition. We show that under certain simplifying assumptions, the decomposition in (3) is exact and models time-varying camera motion and object geometry as two terms. We consider a simple case of a rigid body motion observed through a perspective camera. For points on the object, we can consider only the relative motion between the camera and the object and attribute it all to the camera for simplicity.

Given (stacked) camera projection matrices $W_t \in \mathbb{R}^{3T\times 4}$, points $\tilde{X}_k \in \mathbb{R}^{4\times N}$ in homogenous coordinates that remain at constant projective depth $\mathbf{d} \in \mathbb{R}^{N\times 1}$ from the camera over the whole sequence, we note the following equation [23]:

$$\tilde{P}_k = W_t \tilde{X}_k \operatorname{diag}(\mathbf{d})^{-1}, \tag{5}$$

where $\tilde{P}_k \in \mathbb{R}^{3T\times N}$ is P_k in homogenous coordinates. Both W_t and $\tilde{X}_k \operatorname{diag}(\mathbf{d})^{-1}$ can be recovered by considering a truncated SVD at rank 4: $W_t = \lfloor U_k \rfloor_4 \lfloor \Sigma_k \rfloor_4$, and $\tilde{X}_k \operatorname{diag}(\mathbf{d})^{-1} = \lfloor V_k \rfloor_4^\top$.

The trajectory matrix factorises into the time-varying camera matrices and object geometry. As the depth is not constant in the real-world setting, this decomposition is approximate and suggests the following alternative loss:

$$\mathcal{L}_{\text{per}} = \sum_k \|\tilde{P}_k - W_t \tilde{X}_k \operatorname{diag}(\mathbf{d})^{-1}\|_F^2, \tag{6}$$

where W_t, and $\tilde{X}_k \operatorname{diag}(\mathbf{d})^{-1}$ are obtained via SVD as above.

Choice of r. Setting r correctly is important. Intuitively, it captures the degrees of freedom present in the trajectory data or the number of trajectories that are sufficient to form a basis. From the analysis above, we saw that rank $r = 4$ corresponds to assuming constant depth and perspective camera. However, higher r is needed to tolerate changing depth and tracking errors [12, 23]. Similarly, not all motion is rigid in real-world videos, which also requires increasing r. We empirically determined $r = 5$ to yield good results.

3.3 Training a segmenter using flow and trajectories

The losses above require optical flow F, trajectories P, and masks M_k obtained using a segmentation network $\Phi(\mathcal{I}) = M$. This suggests a simple procedure of training a segmentation network given a dataset of videos, which we summarise in Fig. 2. We precalculate optical flow for each frame and obtain a set of point trajectories for each video using off-the-shelf pretrained networks. For training, we consider triples of $(\mathcal{I}, F, P)_i$ for each frame i, where for trajectories P, we take trajectories for which the points are visible in the image $\mathcal{I}$. This can be accomplished by making use of visibility predictions in the output of point trackers or calculating trajectories by querying points in each frame. We use bilinear sampling for $\pi(\cdot)$ to obtain mask values at trajectory coordinates.

Temporal smoothing. We include a temporal smoothing loss, which matches mask predictions between two frames offset by Δt using the predicted trajectories:

$$\mathcal{L}_\tau = \|\pi(\Phi(\mathcal{I}_t), (P_t)_t) - \pi(\Phi(\mathcal{I}_{t+\Delta t}), (P_t)_{t+\Delta t})\|_2^2, \tag{7}$$

where $\mathcal{I}_t$ is the t-th frame and P_t are trajectories associated with t-th frame. We write the final loss as: $\mathcal{L} = \lambda_f \mathcal{L}_f + \lambda_t \mathcal{L}_t + \lambda_\tau \mathcal{L}_\tau$, where $\lambda_f, \lambda_t, \lambda_\tau$ balance the contribution of the different loss terms.

https://doi.org/10.52202/079017-3576

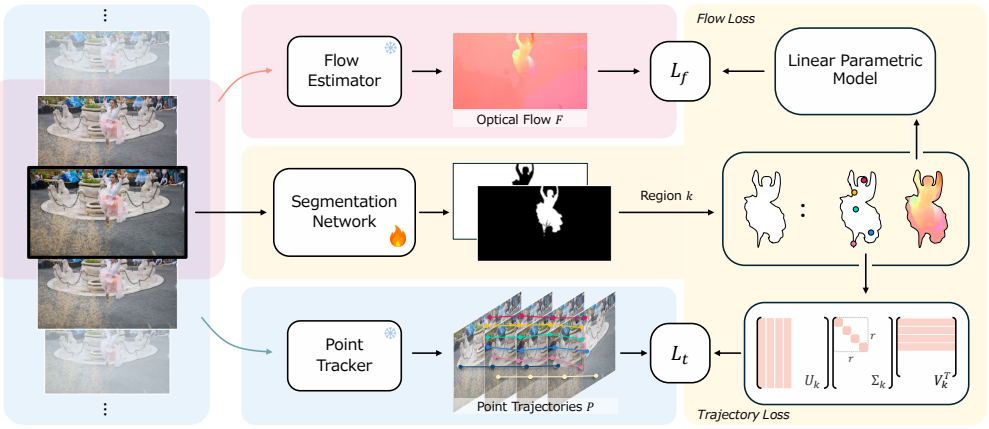


Figure 2: **Overview of our approach.** We self-supervise a segmentation network, i.e., without access to mask annotations, using both short-term motion information (optical flow) and long-term motion (point trajectories). We design a loss function that encourages the segmentation network to cluster regions where trajectories form low-rank-r groups, which should align well with objects. Off-the-shelf methods are used to estimate optical flow and point trajectories given a dataset of videos.

Choice of k. Following prior work [10], we set k, the number of predicted masks, to be higher than the maximum number of objects in the scene to account for potential parallax and non-rigid motion. In the binary segmentation case, we recover two components by considering the average appearance feature of each component and solving for the normalised cut on a graph with k nodes.

4 Feasibility study

Our proposed trajectory loss (4) enables training a segmentation network using trajectory data. We first show the feasibility of the proposed cost function in a controlled setting, without actually training Φ. To this end, we consider a synthetic scene from the MOVI-F Kubric [20] dataset for which we obtain ground-truth trajectories for every point and ground-truth object segmentation masks. We explore the loss landscape of the proposed formulation by corrupting the segmentation masks along several principled axes and studying the effect of such corruptions on the trajectory loss.

First, we consider a random alteration of mask pixels, which we refer to as *mask noise*. We control the amount of mask noise using η such that 0.0 corresponds to no pixels changed and 1.0 corresponds to completely random masks. Along this axis, we test whether our loss favours predictions with lower noise. Second, we consider structural alterations, namely under/over-segmentation. To simulate under-segmentation, we merge object masks with the background at random. To simulate over-segmentation, we randomly split the existing object mask into two parts in the middle along either the x or y-axis. We represent this type of mask corruption using integers. Negative values indicate the number of objects removed, while positive values correspond to new objects generated from existing ones. Such structural corruption investigates whether the loss can correctly identify the number of moving objects. Finally, we consider the "softness" of the predicted masks by transforming masks into logits and increasing the temperature τ in the softmax operation. This tests whether the loss will prefer low-entropy values. We leave further details of the corruption procedure to Appendix D.

The results of these analyses are shown in Figure 3. All three plots show the loss value as a function of structural corruption. The trajectory loss decreases as the noise and temperature of the masks are reduced, as seen in the first two plots. The third plot also shows that such solutions are preferred in combination. Furthermore, we observe that the loss values are lower when the correct number of segments is detected, and this holds even in the presence of noise or when masks are more uniform. Note, however, that over-segmentation is penalised less than under-segmentation, i.e., missing moving objects leads to a higher value of the loss than, e.g., splitting an object into several components.

https://doi.org/10.52202/079017-3576

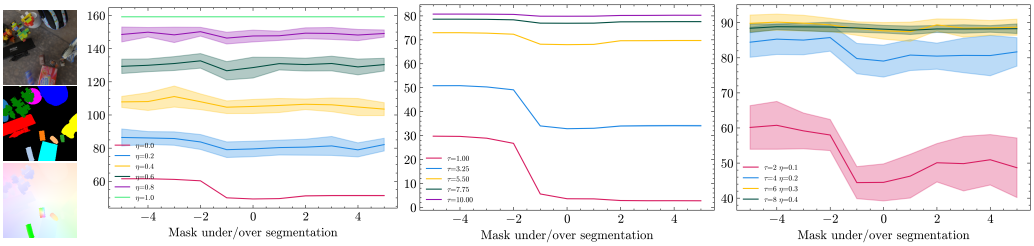


Figure 3: **Feasibility analysis of $\mathcal{L}_t$.** Using a synthetic sequence (left), we vary the amount of noise η injected into the mask, the temperature τ of the mask logits and plot the loss value as a function of the mask under/over segmentation. The plots show that the loss is reduced in low-noise, low-entropy settings and penalises both over- and under-segmentation.

5 Experiments

In this section, we evaluate our approach for unsupervised motion segmentation and compare it with simple baselines and prior subspace clustering methods. Next, we compare our method with state-of-the-art methods for unsupervised video object segmentation across several datasets in a binary segmentation setting. We finish with ablation experiments of our approach.

Datasets. We consider four primary datasets in this study. We use the synthetic MOVi-F variant of the Kubric [20] dataset with ground truth trajectories for comparison with subspace clustering-based approaches. We adopt this setting to eliminate noise in point trajectories as previous methods are sensitive to it. We report the adjusted Rand index (ARI) as the main metric, measuring how close clustering is to the ground truth up to the permutation of cluster identities, where 1 is a perfect match, and 0 means roughly random assignment. We also report FG-ARI, i.e., ARI only on foreground pixels (determined by ground truth masks), which identifies how well different objects are separated.

We also evaluate our approach on real-world datasets: DAVIS 2016 [53], SegTrackv2 (STv2) [35], and FBMS [51], which are popular benchmarks for video object segmentation. Following standard practice [69, 70], foreground objects in STv2 and FBMS are consolidated. We report the Jaccard ($\mathcal{J}$) score, computed using Hungarian matching between predicted and ground truth segmentations.

Implementation. For the experiments on real-world datasets, optical flow is estimated using RAFT [60] and point trajectories using CoTracker [28]. Trajectories are computed within a context window $f = 20$ around each frame, with reflection padding around video boundaries, resulting in chunks of $T = 2f + 1 = 41$ frames. To reduce the effect of noisy predictions, we also filter trajectories along the time dimension using an average filter with a window size of 11. For the experiments on MOVi-F, a small U-Net [56] is trained as the segmentation network, starting from random initialisation. For fairness of comparisons on DAVIS, STv2 and FBMS, we use the same architecture as in [10] — MaskFormer with DINO backbone. We specify further details in Appendix E.

Table 1: Comparison of our LRTL trajectory-based formulation with prior methods.

	MOVi-F	
Method	ARI↑	FG-ARI↑
K-Means	15.26	42.53
SSC [17]	11.12	39.21
LRR [38]	7.47	37.36
LRTL (Ours)	46.07	65.76

5.1 Comparison to trajectory-based methods

In Table 1, we compare our low-rank trajectory loss (LRTL) with prior subspace clustering approaches in a per-video optimisation setting. Subspace clustering operates on a similar intuition to our proposed trajectory loss by a grouping of trajectories that should be linearly dependent. We also consider K-means clustering of trajectories as a simple baseline. For fair comparisons, we train our segmentation model optimising *only* the trajectory loss ($\mathcal{L}_t$). We use $k = 25$ components for each video and train for 5000 steps. This is comparable to the computation requirements and steps of other methods. For K-means, SSC [17] and LRR [38], we search for an optimal set of hyperparameters and the number of components k, reporting the best results. Our approach shows significantly stronger performance than simple K-Means and subspace clustering approaches.

https://doi.org/10.52202/079017-3576

Table 2: **Unsupervised video segmentation** on DAVIS, SegTrackv2, and FBMS. Where possible, we report results without widely applicable post-processing (e.g., CRF) or indicate results in grey.

Method	Inf. Input RGB	Inf. Input Motion	Input Resolution	Motion Est. Method	DAVIS $\mathcal{J}\uparrow$	STv2 $\mathcal{J}\uparrow$	FBMS $\mathcal{J}\uparrow$
Single-sequence methods							
FTS [52]	✓	✓	–	LDOF [5]	55.8	47.8	47.7
CUT [31]	✓	✓	–	LDOF [5]	55.2	54.3	57.2
DS [72]	✓	✓	240 × 426	RAFT [60]	79.1	72.1	71.8
Ponimatkin et al. [54]	✓	✗	480 × 848	ARFlow [39]	80.2	74.9	70.0
OCLR [67] (test ft.)	✓	✓	480 × 848	RAFT [60]	80.9	72.3	69.8
Single-stage end-to-end methods							
OCLR [67]	✗	✓	112 × 224	RAFT [60]	72.1	67.6	65.4
DivA [34]	✓	✓	128 × 224	RAFT [60]	72.4	64.6	60.9
Meunier et al. [45]	✗	✓	128 × 224	RAFT [60]	73.2	55.0	–
GWM [10]	✓	✗	128 × 224	RAFT [60]	79.5	78.9	78.4
Multi-stage methods							
RCF [37]	✓	✗	480 × 848	RAFT [60]	80.9	76.7	69.9
LOCATE [59]	✓	✗	480 × 848	ARFlow [39]	80.9	79.9	68.8
LRTL (Ours)	✓	✗	192 × 352	RAFT [60] CoTracker [28]	**82.2**	**81.2**	**79.6**

Table 3: **Alternative losses** to our proposal. Other variants do not match the performance of our formulation.

Loss	DAVIS ($\mathcal{J}\uparrow$)
$\mathcal{L}_{\text{rec@3}}$ (3)	11.1
$\mathcal{L}_{\text{per}}$ (6)	18.2
$\mathcal{L}_{\text{rec@5}}$ (3)	14.6
tracks-as-flow	65.3
Ours $\mathcal{L}_t$ (4)	71.9

Table 4: **Ablation of loss terms.** All loss terms synergise to improve performance.

Loss	DAVIS ($\mathcal{J}\uparrow$)
$\lambda_f\mathcal{L}_f$	78.5
$\lambda_t\mathcal{L}_t$	71.9
$\lambda_t\mathcal{L}_f + \lambda_t\mathcal{L}_t$	81.7
$\lambda_t\mathcal{L}_f + \lambda_t\mathcal{L}_t + \lambda_\tau\mathcal{L}_\tau$	82.2

5.2 Unsupervised video object segmentation

We compare to recent methods on the unsupervised video object segmentation task *without first-frame prompting or post-processing*. In this setting, we train a single network on the benchmark datasets for binary video segmentation. We compare with *single-sequence methods* that perform optimisation for each sequence/video individually. Additionally, we benchmark dataset-wide *single-stage end-to-end methods* where training is performed over multiple videos simultaneously, training a network in an end-to-end manner. We also compare with *multi-stage methods* that train and re-train several networks. We report our results on standard benchmarks in Table 2. While the closest prior work relies on multiple stages of training, pseudo-labelling, applying CRF, and retraining, our end-to-end trained method shows better performance at lower resolutions. We attribute this to the effectiveness of our approach in incorporating long-term motion information.

In Fig. 4, we show qualitative results of our approach and compare with RCF [37], a state-of-the-art multi-stage approach. Our network trained with both flow and trajectory losses yields segmentations with noticeably better boundaries despite operating at a lower resolution. Notably, our formulation also effectively avoids segmenting shadows and water ripples of the swan, which are difficult to separate based on instantaneous motion alone.

5.3 Ablations

Alternative losses. We have explored several alternative formulations of the trajectory loss in our approach and present the analysis in Table 3. Losses based on full SVD reconstruction fail to train a network sufficiently. $\mathcal{L}_{\text{per}}$ performs the best out of these, likely as DAVIS contains several scenes

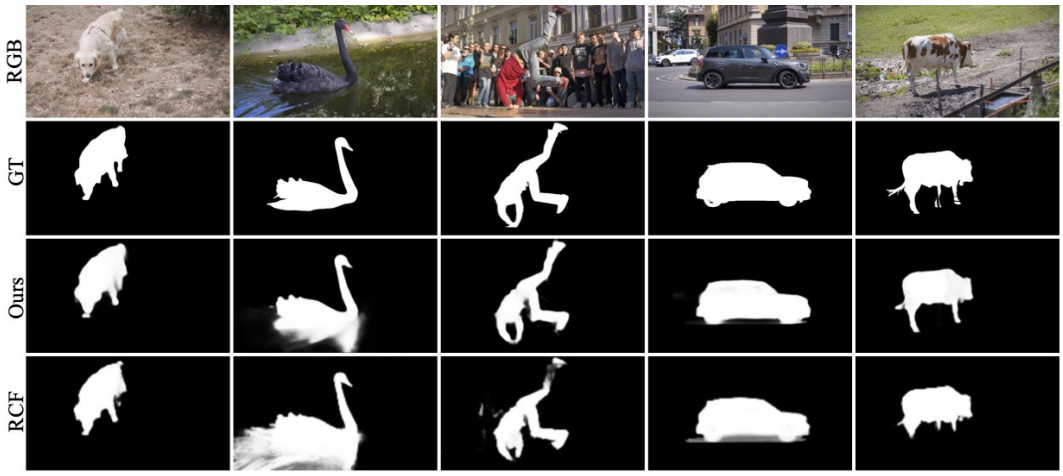


Figure 4: Qualitative comparison of our results on DAVIS with RCF which uses higher resolution and multi-stage training. Our method contains slightly better boundaries, does not segment shadows and separates water ripples from the swan.

with a panning camera tracking a rigid object at an approximately constant distance, which matches the assumptions. Increasing or decreasing the rank of the approximation performs worse. We also consider *track-as-flow* loss, where trajectories P are treated as optical flow by subtracting positions from adjacent times. Then, for T frames, Eq. (2) can be applied. We find that such a formulation still underperforms in comparison to our trajectory-based formulation (Eq. (4)).

We believe our formulation provides better results than the above for two possible reasons. First, by minimising higher-than-r singular values, we are not *strictly* enforcing assumptions like rigidity. Second, our loss formulation is more numerically stable as it requires only gradients w.r.t. to the singular values. As we seek to drive them close to zero, the matrices P_k become increasingly ill-conditioned as the training progresses. Additionally, gradients w.r.t. U and $V^\top$ depend on inverse singular values Σ^{-1} [63], which become numerically unstable as they are approaching zero. On the other hand, $\mathrm{d}\,\Sigma = I_N \circ (U^\top \,\mathrm{d}\,P_k\, V)$ does not have this problem.

Influence of losses. In Table 4, we consider the method with only the flow loss component and only the trajectory loss component. We find that our trajectory-based loss improves flow-only performance. Using only trajectory-based loss shows weaker performance than just optical flow, likely due to only a sparse set of points and noise introduced by estimating positions for occluded points. Ablating temporal smoothing loss slightly lowers performance as well.

Limitations. While we have demonstrated the effectiveness of learning segmentation from long-term motion, there is potential for further improvements in leveraging point trajectories. First, while modern trackers predict reasonable positions for occluded points, naturally, these predictions are less accurate. Thus, a more explicit handling of occlusions and tracking noise would likely help. Second, we currently only use trajectory estimates from nearby frames for training. This means that we sometimes track the same point multiple times, which could be avoided with caching trajectories. While we handle non-rigidity using over-segmentation, extending this principle to video with multiple non-rigid objects is an important feature direction.

6 Conclusion

We have introduced a principled method to train an image segmentation network using long-term motion information expressed as point trajectories. Our trajectory loss formulation follows the principle of common fate and aims to group trajectories into low-rank matrices, representing the idea the motion of points belonging to the same object can be roughly explained as a combination of other points. Using synthetic data we have shown that such a loss should prefer low-noise and low-entropy solutions as well as identify the correct number of moving objects. In comparison with other methods,

https://doi.org/10.52202/079017-3576

our loss formulation has shown superior performance compared to subspace clustering baselines on synthetic data and achieved state-of-the-art results on unsupervised video object segmentation benchmarks when combined with optical flow-based loss.

Acknowledgements L. K. is supported by is supported by EPSRC AIMS CDT EP/S024050/1. I. L., C. R. and A. V. are supported by ERC-CoG UNION 101001212 and EPSRC VisualAI EP/T028572/1.

References

[1] Gilad Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE transactions on pattern analysis and machine intelligence*, (4):384–401, 1985. 4

[2] Görkay Aydemir, Weidi Xie, and Fatma Guney. Self-supervised object-centric learning for videos. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[3] Pia Bideau and Erik Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *European Conference on Computer Vision*, pages 433–449. Springer, 2016. 2

[4] Pia Bideau, Aruni RoyChowdhury, Rakesh R Menon, and Erik Learned-Miller. The best of both worlds: Combining cnns and geometric constraints for hierarchical motion segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 508–517, 2018. 2

[5] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2010. 8

[6] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V 11*, pages 282–295. Springer, 2010. 3

[7] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017. 2

[8] Jason Chang and John W. Fisher III. Topology-constrained layered tracking with latent flow. *2013 IEEE International Conference on Computer Vision*, pages 161–168, 2013. 2

[9] Anil M Cheriyadat and Richard J Radke. Non-negative matrix factorization of partial track data for motion segmentation. In *2009 ieee 12th international conference on computer vision*, pages 865–872. IEEE, 2009. 3

[10] Subhabrata Choudhury, Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion. In *British Machine Vision Conference (BMVC)*, 2022. 2, 3, 4, 6, 7, 8, 14, 17, 18

[11] Joao Costeira and Takeo Kanade. A multi-body factorization method for motion analysis. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1071–1076. IEEE, 1995. 3

[12] Joao Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29:159–179, 1998. 5

[13] Shuangrui Ding, Weidi Xie, Yabo Chen, Rui Qian, Xiaopeng Zhang, Hongkai Xiong, and Qi Tian. Motion-inductive self-supervised object discovery in videos. *arXiv preprint arXiv:2210.00221*, 2022. 2

[14] Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens Continente, Kucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. In *NeurIPS Datasets Track*, 2022. 1, 3

[15] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. *arXiv preprint arXiv:2306.08637*, 2023. 1, 3, 14, 16

[16] Carl Doersch, Yi Yang, Dilara Gokay, Pauline Luc, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ross Goroshin, João Carreira, and Andrew Zisserman. Bootstap: Bootstrapped training for tracking-any-point. *arXiv preprint arXiv:2402.00847*, 2024. 14

[17] Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013. 2, 3, 7, 18

[18] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014. 2

[19] Matthieu Fradet, Philippe Robert, and Patrick Pérez. Clustering point trajectories with various life-spans. In *2009 Conference for Visual Media Production*, pages 7–14. IEEE, 2009. 3

[20] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3761, 2022. 6, 7, 16, 17, 18

[21] Benjamin David Haeffele, Chong You, and Rene Vidal. A critique of self-expressive deep subspace clustering. In *International Conference on Learning Representations*, 2020. 3

[22] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pages 59–75. Springer, 2022. 1, 3, 16

[23] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 5

[24] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *European Conference on Computer Vision*, pages 668–685. Springer, 2022. 1

[25] Suyog Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. *arXiv preprint arXiv:1701.05384*, 2017. 2

[26] Allan D. Jepson and Michael J. Black. Mixture models for optical flow computation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–761, 1993. 2

[27] N. Jojic and B.J. Frey. Learning flexible sprites in video layers. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001. 2

[28] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 1, 3, 7, 8, 14, 16, 17

[29] Laurynas Karazija, Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised multi-object segmentation by predicting probable motion patterns. *Advances in Neural Information Processing Systems*, 35:2128–2141, 2022. 2

[30] Margret Keuper. Higher-order minimum cost lifted multicuts for motion segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4242–4250, 2017. 3

[31] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the IEEE international conference on computer vision*, pages 3271–3279, 2015. 3, 8

[32] Margret Keuper, Siyu Tang, Yu Zhongjie, Bjoern Andres, Thomas Brox, and Bernt Schiele. A multi-cut formulation for joint segmentation and tracking of multiple objects. *arXiv preprint arXiv:1607.06317*, 2016. 2

[33] Hala Lamdouar, Weidi Xie, and Andrew Zisserman. Segmenting invisible moving objects. In *BMVC*, 2021. 2

[34] Dong Lao, Zhengyang Hu, Francesco Locatello, Yanchao Yang, and Stefano Soatto. Divided attention: Unsupervised multi-object discovery with contextually separated slots. *arXiv preprint arXiv:2304.01430*, 2023. 2, 8

[35] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. *2013 IEEE International Conference on Computer Vision*, pages 2192–2199, 2013. 2, 7

[36] Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C-C Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6526–6535, 2018. 2

 https://doi.org/10.52202/079017-3576

[37] Long Lian, Zhirong Wu, and Stella X Yu. Bootstrapping objectness from videos by relaxed common fate and visual grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14582–14591, 2023. 2, 8, 15

[38] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35 (1):171–184, 2012. 2, 3, 7

[39] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 8

[40] Runtao Liu, Zhirong Wu, Stella Yu, and Stephen Lin. The emergence of objectness: Learning zero-shot segmentation from videos. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[41] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pages 347–360. Springer, 2012. 3, 18

[42] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3623–3632, 2019. 2

[43] Dijun Luo, Feiping Nie, Chris Ding, and Heng Huang. Multi-subspace representation and discovery. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II 22*, pages 405–420. Springer, 2011. 3

[44] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 17

[45] Etienne Meunier and Patrick Bouthemy. Unsupervised space-time network for temporally-consistent segmentation of multiple motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22139–22148, June 2023. 8

[46] Etienne Meunier and Patrick Bouthemy. Unsupervised motion segmentation in one go: Smooth long-term model over a video. *arXiv preprint arXiv:2310.01040*, 2023. 2

[47] Etienne Meunier, Anaïs Badoual, and Patrick Bouthemy. Em-driven unsupervised learning for efficient motion segmentation. *CoRR*, abs/2201.02074, 2022. 2

[48] Peter Ochs and Thomas Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *2011 international conference on computer vision*, pages 1583–1590. IEEE, 2011. 2

[49] Peter Ochs and Thomas Brox. Higher order motion models and spectral clustering. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 614–621. IEEE, 2012. 3

[50] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1187–1200, 2013. 3

[51] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1187–1200, 2014. 2, 7

[52] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013. 2, 8

[53] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 2, 7

[54] Georgy Ponimatkin, Nermin Samet, Yang Xiao, Yuming Du, Renaud Marlet, and Vincent Lepetit. A simple and powerful global optimization for unsupervised video object segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5892–5903, 2023. 8

[55] Shankar R Rao, Roberto Tron, René Vidal, and Yi Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 3

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 7, 18

[57] Sadra Safadoust and Fatma Güney. Multi-object discovery by low-dimensional object motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 734–744, 2023. 2

[58] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. *Advances in Neural Information Processing Systems*, 35:18181–18196, 2022. 2

[59] Silky Singh, Shripad Deshmukh, Mausoom Sarkar, and Balaji Krishnamurthy. Locate: Self-supervised object discovery via flow-guided graph-cut and bootstrapped self-training. In *British Machine Vision Conference (BMVC)*, 2023. 2, 8

[60] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 7, 8, 17

[61] Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari. Learning to segment moving objects. *Int. J. Comput. Vision*, 127(3):282–301, mar 2019. ISSN 0920-5691. 2

[62] Philip H. S. Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356:1321 – 1340, 1998. 2

[63] James Townsend. Differentiating the singular value decomposition. 2016. URL https://j-towns.github.io/papers/svd-derivative.pdf. 9

[64] René Vidal and Paolo Favaro. Low rank subspace clustering (lrsc). *Pattern Recognition Letters*, 43:47–61, 2014. 3

[65] Xudong Wang, Ishan Misra, Ziyun Zeng, Rohit Girdhar, and Trevor Darrell. Videocutler: Surprisingly simple unsupervised video instance segmentation. *arXiv preprint arXiv:2308.14710*, 2023. 2, 3, 15

[66] Max Wertheimer. Experimentelle studien uber das sehen von bewegung. *Zeitschrift fur psychologie*, 61: 161–165, 1912. 1

[67] Jun Xie, Weidi Xie, and Andrew Zisserman. Segmenting moving objects via an object-centric layered representation. *ArXiv*, abs/2207.02206, 2022. 2, 8

[68] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV 9*, pages 94–106. Springer, 2006. 3

[69] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7177–7188, 2021. 2, 7, 17

[70] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 7

[71] Yanchao Yang, Brian Lai, and Stefano Soatto. Dystab: Unsupervised object segmentation via dynamic-static bootstrapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2826–2836, 2021. 2

[72] Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. Deformable sprites for unsupervised video decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2657–2666, June 2022. 8

[73] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 15

[74] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023. 14

 https://doi.org/10.52202/079017-3576

Table 5: **Influence of** r, the rank of the trajectory matrix used in loss function (4).

r	**DAVIS** ($\mathcal{J} \uparrow$)
3	76.0
4	79.6
5	82.2
6	80.9

Table 6: **Influence of** k, the number of predicted components before merging.

k	**DAVIS** $\mathcal{J} \uparrow$
2	78.0
3	82.0
4	82.2
5	72.8

Table 7: **Influence of context length** of the trajectory matrix.

Context length	**DAVIS** ($\mathcal{J} \uparrow$)
10	79.1
15	81.0
20	82.2
30	80.8

Table 8: **Influence of trackers** used to estimate point trajectories.

Tracker	**DAVIS** ($\mathcal{J} \uparrow$)
TAPIR [15]	73.4
PIPs++ [74]	74.9
BootsTap [16]	76.8
CoTracker [28]	78.9

Supplementary material

In this supplementary material, we consider additional ablations of our approach (Appendix B), include further results (Appendix C), and provide the implementation details (Appendix E). Accompanying this supplementary material, we include videos of our results on DAVIS and SegTrackv2 datasets. We also include an example video visualising a sample of trajectories that the model receives as input. The code and models will be released upon acceptance.

A Broader impact

Segmentation is a component in a very large and diverse spectrum of applications in healthcare, image processing, computer graphics, surveillance and more. As with many technologies, the application can be good or bad. In this paper, we explore how to train a model to perform segmentation in an unsupervised manner. This has the positive benefit of removing manual labour requirements to obtain annotations, which might also eventually apply to bad actors. We, however, consider the immediate real-world impact beyond the research community of our work here limited as unsupervised systems still show lower performance than supervised counterparts.

B Additional ablations

Rank r of trajectory matrix. In Table 5, we vary r, the rank of the trajectory matrix used in the trajectory loss (Eq. (4)). As previously mentioned, the choice of rank reflects the degrees of freedom in the system and controls implicitly the assumptions about the types of motion and cameras used to capture sequences. At $r = 3$ and 4, we observe slight impact on the performance in comparison to $r = 5$. $r = 5$ appears to be the optimal setting, which is what we used in our main experiments. At $r = 6$, the performance drops again, likely as it becomes sufficient to group and explain simple motions together.

Number of segments k. In Table 6, we vary k, the number of masks predicted by our method, before merging. As in prior work [10], the $k = 4$ appears to be the optimal setting. The performance drops beyond this point as it becomes difficult to group objects.

Influence of context window length T. In Table 7, we vary the length of the context windows (f) and thus T for our method when considering trajectories. We find increasing the context window helps slightly. However, the performance starts to drop afterwards. We hypothesise that this is due to difficulty predicting sensible point trajectories for points that move outside of the frame and become invisible, as DAVIS contains many videos where the camera tracks the main subject. Though several values of this setting are viable.

Table 9: **Alternative network architectures** for segmentation.

Network	**DAVIS** ($\mathcal{J}$ ↑)
UNet	80.6
MaskFormer + Swin-Tiny	81.2
MaskFormer + DINO	82.2

Table 10: **Comparison with appearance-only methods.**

Method	**DAVIS** ($\mathcal{J}$ ↑)
VideoCutLER [65]	67.2
VideoSAUR [73]	17.5
LRTL (Ours)	82.2

Source of tracks. In Table 8, we experiment with different trackers to obtain tracks. We consider TAPIR[1], PIP++[2] and BootsTap[3] along CoTracker. Due to the limitation of some options (PIPs++ not predicting visibility) and inherent noise in invisible tracks for TAPIR, we lowered the context window to 15. We also do not consider tracks from adjacent frames as this seems to lower performance for other trackers. Finally we did not use EMA in these experiments. We observe that CoTracker performs the best while other trackers show slightly weaker results. We hypothesise this is due to CoTracker estimating reasonable trajectories for occluded points, which are included in the matrix P_k. Some trackers, e.g., TAPIR, are restricted to predicting points within the frame, thus providing extremely noisy estimates in scenes where objects move outside the frame.

Alternative networks. As our proposed loss function is network-architecture agnostic as it only requires mask prediction. Thus, any network which predicts masks or has mask-like representation could be used. In Table 9, we experiment with changing the segmenter architecture in the DAVIS benchmark. This shows that we can swap different network architectures with relative ease and obtain similar results.

Inference speed. Here we provide the inference time comparison using different networks as average FPS during DAVIS evaluation. For MaskFormer + DINO configuration, we measure 3.3 FPS, while with UNet we measure 6.4 FPS. Note that since our contribution is a loss function, it is network architecture agnostic. Using it does not affect inference time; only the choice of network architecture does. We matched the architecture with prior work for the best comparisons.

Comparison with appearance-only works. Finally, we include a comparison to unsupervised methods that consider only appearance during learning. In Table 10, we provide a comparison of VideoCutLER [65] and VideoSAUR [73] on DAVIS using the same merging strategy for combining multiple predictions to a binary segmentation as in our method.

Our method shows a significant advantage. We observe that VideoCutLER has trouble segmenting instances from crowds in the background. VideoSAUR has imprecise object boundaries which severely impacts performance when measurred using Jaccard score.

C Additional results

C.1 Qualitative results on SegTrackv2

In Fig. 5, we provide additional qualitative results from our approach on the SegTrackv2 dataset. We compare with the state-of-the-art multi-stage Relaxed Common Fate (RCF) approach [37]. Our method correctly identifies more parts of the objects and has better boundaries.

D Parametric mask alterations

In this section, we show the effect of the parametric ground truth mask alterations used to study the trajectory loss in section 4.1. The purpose of these alterations is to disturb ground truth masks in a

[1]Code and models available https://github.com/google-deepmind/tapnet under Apache-2.0 license.

[2]Code and models available https://github.com/aharley/pips2 under MIT license.

[3]Code and models available https://github.com/google-deepmind/tapnet under Apache-2.0 license.

https://doi.org/10.52202/079017-3576

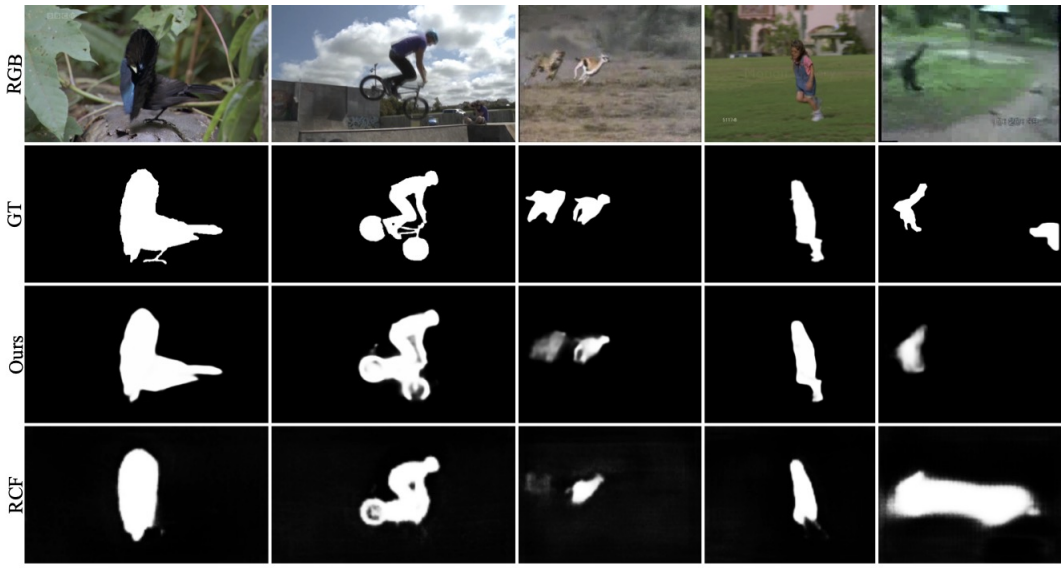


Figure 5: Qualitative comparison of our results on SegTrackv2 with RCF which uses higher resolution and multi-stage training. Our method contains slightly better boundaries and segments more whole objects.

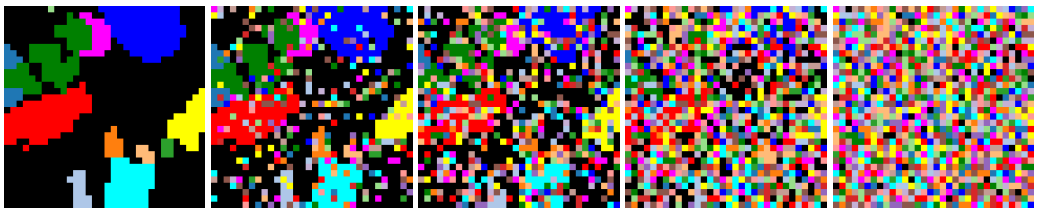


Figure 6: Example *noise* mask alteration. The parameter η is the probability of assigning a mask pixel at random.

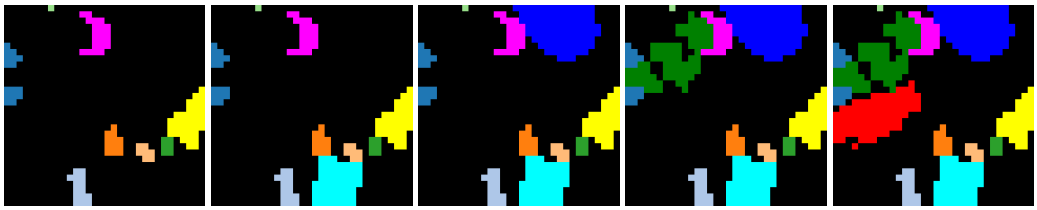
controlled way to enable studying the effect this has on the loss. For this purpose, we use synthetic data from MOVi-F sequences of the Kubric [20] dataset suite, which is the same data that is used to train CoTracker [28], TAPIR [15], PIP [22] and similar. We consider three types of alterations:

- The first kind of alteration is random *noise*. With probability η, we set each mask pixel to a random class sampled from $\mathcal{U}(0, K)$, where $K = 20$ in this case. When $\eta = 0$, thus, there is no alteration. When $\eta = 0.5$, around half of the mask pixels (in expectation) are assigned randomly. Fig. 6 shows the effect of η in practice.

- The second kind of alteration we consider is a *structural* change meant to approximate over/under-segmentation. For under-segmentation, we change the mask regions corresponding to the whole object to the background. Fig. 8 shows this in effect. For over-segmentation, we split an existing component randomly along an axis passing through the object centre and parallel to either the x- or y-axis at random. Fig. 9 shows this in effect. We parameterise this alteration with integers s, where a positive number controls the number of components split, and negative numbers correspond to the number of components set to the background.

- The third kind of alteration is *temperature*. Its purpose is to model how the entropy of the categorical distribution modelled by the segmentation network might affect the loss. For this, we increase the temperature τ in softmax operation $softmax(l/\tau)$ for logits l calculated from the input mask, which results in increasingly "soft" masks.

https://doi.org/10.52202/079017-3576



$s = -4$ $s = -3$ Figure 7: $s = -2$ $s = -15$ $s = 0$

Figure 8: Example mask *structural* mask alteration showing modification used to approximate under-segmentation. The parameter s controls the number of objects set to the background.

$s = 0$ $s = 1$ $s = 2$ $s = 3$ $s = 4$

Figure 9: Example *structural* mask alteration modelling over-segmentation. The parameter s controls the number if objects split into two at random.

The three types of alteration are composed to generate a synthetic prediction mask that can be used to investigate how the trajectory loss behaves as the mask changes. We use 25 trials to estimate the value of the loss for a given configuration of the parameters s, η, τ.

E Implementation details

Here, we further specify the configuration and implementation details used in our experiments.

E.1 Extracting flow

We use RAFT [60][4] to extract optical flow pretrained on FlyingThings3D [44]. We follow the methods used to extract flow in previous work [10, 69]. Namely, we consider pairs of frames with a distance in time of either 1 or 2, both in forward and backward directions for DAVIS and SegTrackv2. For FBMS, we consider distances of 3 and 6 due to lower motion setting in the dataset. The optical flow is extracted before training.

E.2 Extracting trajectories

We use CoTracker [28] to extract point trajectories. CoTracker is trained on MOVi-F Kubric [20] datasets. We use CoTracker v2[5]. We query at every 4th-pixel coordinate for each frame to extract point trajectories. At 480×854 resolution for DAVIS, this results in 25k points for each frame. When tracking, we find it beneficial to inject auxiliary query points. For this, we define two additional query grids with a stride of 32, querying a frame seven frames in the future and the past (or less if at the video boundaries). This generates around 2k additional points, which we do not use for training. When processing videos of heterogeneous resolutions, we resize the input to 480×854 to maintain the same number of points.

[4]Code and models available https://github.com/princeton-vl/RAFT/tree/master under BSD-3 license.

[5]Code and models available https://github.com/facebookresearch/co-tracker under non-comercial license.

https://doi.org/10.52202/079017-3576

E.3 Training hyperparameters

For the segmentation network, we use the same model architecture as in [10] – MaskFormer with DINO backbone. We feed images at 192×352 resolution. We also use random horizontal flipping augmentation. The network is trained to predict $k = 4$ components, which, in the case of binary segmentation, are then merged into two following [10]. We train using AdamW optimiser, with a learning rate of 1.5e-4, weight decay of 0.01, a batch size of 8, and a linear learning warmup schedule for 1500 iterations. We train for 5000 iterations.[6] We use an Exponential Moving Average (EMA) with the decay power of 2/3 with a warmup of 1500 iterations and update every 10 steps to help stabilise the training. On SegTrackv2 we instead used decay power 4/5 as the dataset is considerably smaller than others. We set $\lambda_f = 0.03$, $\lambda_t = 5 * 10^-5$, and $\lambda_\tau = 0.1$ in all experiments, which yields loss values in a similar numerical range. For the temporal smoothing loss, we use $\Delta t = 5$.

E.4 MOVi-F experiments

When conducting experiments on the MOVi-F dataset (Sec. 4.2), we consider ground-truth trajectories obtained from modified rendering script [20]. We normalise the trajectories to the $[0, 1]$ range based on image width and height.

For K-Means, we consider the trajectories with the initial position at $T = 0$ subtracted, thus clustering offsets from the initial position.

For SSC [17], we translate the method to Python following the original implementation in Matlab[7]. We use the ADMM variant, which we found to give better results. We set the $\alpha = 100$ and kept the rest of the hyperparameters unchanged. To transform the coefficient matrix into a graph adjacency, we found that simple symmetrisation yielded slightly better results than the proposed method that additionally normalised and filtered values. We report results for this method using the optimal number of clusters for spectral clustering.

For LRR [41], we similarly translate the method to Python following the original implementation in Matlab[8]. We use $\lambda = 0.2$. Additionally, we found it beneficial to reduce $\rho = 1.01$ and use a larger number of iterations (10k) than proposed. Similarly to SSC, we experimented with different ways to transform the coefficient matrix to adjacency, including automatically determining the number of clusters based on the block-diagonal structure. We found, however, that using simpler symmetrisation with optimal numbers of clusters determined by an oracle gave the best results.

When considering our trajectory loss, we parameterise the masks with a small randomly initialised Unet [56] predicting a 25-way segmentation, which we optimize using AdamW optimizer.

Note that K-Means, SSC and LRR baselines cluster trajectories rather than segmenting the image. To map back to the image domain and obtain segmentation masks, we repeatedly apply the method for each frame within a sequence, considering the trajectory for each pixel. This enables the most direct way to establish segmentation of the images through significant additional computation effort. An alternative could be to consider sequence wide-trajectories jointly; however, approaches like SSC and LLR do not scale well to such a large number of trajectories. For our trajectory loss, optimisation can be performed per sequence and, as we show in our real-world experiments, dataset-wide.

[6] We estimate about 3 hours to train a model using A6000 GPU (peak GPU memory 25GB). We estimate around 100 GPU hours to train models for the results here.

[7] Code available at http://www.vision.jhu.edu/code/

[8] Code available at https://sites.google.com/site/guangcanliu/

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have confirmed the viability of our loss formulation in controlled simulated settings, per-sequence optimisation settings with no tracking noise and in real-world settings. We also considered alternative formulations of the trajectories and found them to underperform.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Section 5.3.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/079017-3576

Answer: [NA]

Justification: the paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We list all relevant details in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not include code at the time of submission but commit to releasing the code and models at a later time.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We include brief summary of key details in Section 5 and complete information in Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We do not include confidence intervals when reporting main experimental results due to the computational burden of doing so. We report $\pm\sigma$ intervals in our feasibility study.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

https://doi.org/10.52202/079017-3576

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We give the computation cost of a single experiment and estimate total GPU hours required to train models for the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We make use of publicly available and open-source code and models, respecting individual licenses.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: As our key proposal is a method for learning segmentation using trajectory data, we do not foresee our models trained for benchmark datasets requiring safeguards as their use is limited due to the small dataset scale.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the models we build upon which are released on open licenses permitting such use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

https://doi.org/10.52202/079017-3576

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-3576