

BubbleNets: Learning to Select the Guidance Frame in Video Object Segmentation by Deep Sorting Frames

Brent A. Griffin Jason J. Corso
 University of Michigan
 {griffb, jjcorso}@umich.edu

Abstract

Semi-supervised video object segmentation has made significant progress on real and challenging videos in recent years. The current paradigm for segmentation methods and benchmark datasets is to segment objects in video provided a single annotation in the first frame. However, we find that segmentation performance across the entire video varies dramatically when selecting an alternative frame for annotation. This paper addresses the problem of learning to suggest the single best frame across the video for user annotation—this is, in fact, never the first frame of video. We achieve this by introducing BubbleNets, a novel deep sorting network that learns to select frames using a performance-based loss function that enables the conversion of expansive amounts of training examples from already existing datasets. Using BubbleNets, we are able to achieve an 11% relative improvement in segmentation performance on the DAVIS benchmark without any changes to the underlying method of segmentation.

1. Introduction

Video object segmentation (VOS), the dense separation of objects in video from background, remains a hotly studied area of video understanding. Motivated by the high cost of densely-annotated user segmentations in video [5, 38], our community is developing many new VOS methods that are regularly evaluated on the benchmark datasets supporting VOS research [22, 31, 33, 37, 45]. Compared to unsupervised VOS [12, 21, 29, 44], semi-supervised VOS, the problem of segmenting objects in video given a single user-annotated frame, has seen rampant advances, even within just the past year [2, 4, 7, 8, 9, 16, 17, 25, 28, 30, 35, 46].

The location and appearance of objects in video can change significantly from frame-to-frame, and, from our own analysis, we find that using different frames for annotation changes performance dramatically, as shown in Figure 1. Annotating video data is an arduous process, so it

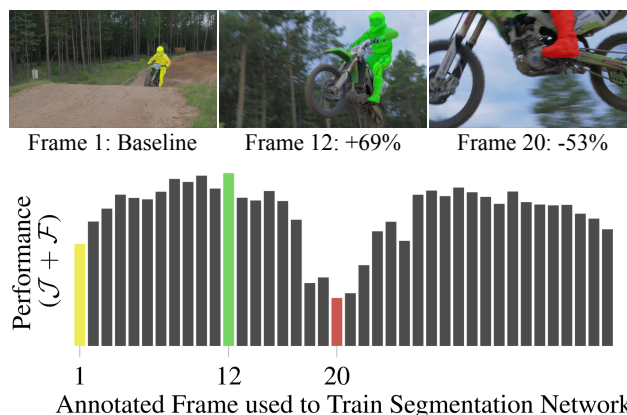


Figure 1. The current paradigm for video object segmentation is to segment an object annotated in the first frame of video (yellow, left). However, selecting a different frame for annotation changes performance across the entire video [for better (green) or worse (red)]. To best use an annotator’s time, our deep sorting framework suggests a frame that will improve segmentation performance.

is critical that we improve performance of semi-supervised VOS methods by providing the best single annotation frame possible. However, we are not aware of any work that seeks to learn which frame to annotate for VOS.

To that end, this paper addresses the problem of selecting a single video frame for annotation that will lead to greater performance. Starting from an untouched video, we select an annotation frame using our deep bubble sorting framework, which makes relative performance predictions between pairs of frames using our custom network, BubbleNets. BubbleNets iteratively compares and swaps adjacent video frames until the frame with the greatest predicted performance is ranked highest, at which point, it is selected for the user to annotate and use for VOS. To train BubbleNets, we use an innovative relative-performance-based loss that increases the number of training examples by orders of magnitude without increasing frame labeling requirements. Finally, we evaluate BubbleNets annotation frame selection on multiple VOS datasets and achieve as much as an 11% relative improvement in combined Jaccard

measure and region contour accuracy ($\mathcal{J}+\mathcal{F}$) over the same segmentation method given first-frame annotations.

The first contribution of our paper is demonstrating the utility of alternative annotation frame selection strategies for VOS. The current paradigm is to annotate an object in the first frame of video and then automatically segment that object in the remaining frames. We provide thorough analysis across four datasets and identify simple frame-selection strategies that are immediately implementable for all VOS methods and lead to better performance than first-frame selection. To the best of our knowledge, this represents the first critical investigation of segmentation performance for different annotation frame selection strategies.

The second contribution of our paper is the deep bubble sorting framework and corresponding implementation that improves VOS performance. We are not aware of a single paper that investigates selection of the annotated frame in VOS. The necessary innovation for our network-based approach is our loss formulation, which allows extensive training on relatively few initial examples. We provide details on generating application-specific performance labels from pre-existing datasets, and our deep sorting formulation is general to all video processes that train on individual frames and have a measurable performance metric. Using our custom network architecture and a modified loss function inspired by our VOS frame-selection analysis, we achieve the best frame-selection-based segmentation performance across all four evaluation datasets.

We provide source code for the current work at <https://github.com/griffbr/BubbleNets> and a video at <https://youtu.be/0kNmm8SBnnU>.

2. Related Work

2.1. Video Object Segmentation

Multiple benchmarks are available to evaluate VOS methods, including: SegTrackv2 [22, 37]; DAVIS 2016, 2017, and 2018 [5, 31, 33]; and YouTube-VOS [45]. Moving away from the single-object hypothesis of DAVIS 2016, these datasets are increasingly focused on the segmentation of multiple objects, which increases the need for a user-provided annotation to specify each object of interest and has led to the development of more semi-supervised VOS methods using an annotated frame. With some exceptions [1, 13, 27, 32], the majority of semi-supervised VOS methods use an artificial neural network.

The amount of training data available for learning-based VOS methods has increased dramatically with the introduction of YouTube-VOS, which contains the most annotated frames of all current VOS benchmarks. However, due to the high cost of user annotation [5, 38], YouTube-VOS only provides annotations for every fifth frame. Operating on the assumption that every frame should be available to the user

for annotation, we obtain training data from, and base the majority of our analysis from, DAVIS 2017, which contains the most training and validation examples of all fully annotated datasets and has many challenging video categories (e.g., occlusions, objects leaving view, appearance change, and multiple interacting objects).

For our BubbleNets implementation that selects annotated frames for VOS, we segment objects using One-Shot Video Object Segmentation (OSVOS) [4], which is state-of-the-art in VOS and has influenced other leading methods [25, 42]. OSVOS uses a base network trained on ImageNet [10] to recognize image features, re-trains on DAVIS 2016 to segment objects in video, and then fine-tunes the network for each video using a user-provided annotation. One unique property of OSVOS is that it does not require temporal consistency, i.e., the order that OSVOS segments frames is inconsequential. Conversely, even when segmentation methods operate sequentially [2, 16, 18, 23, 28, 30, 46], segmentation can propagate forward *and backward* from annotated frames selected later in a video.

2.2. Active Learning

Active learning (AL) is an area of research enabling learning algorithms to perform better with less training by letting them choose their own training data. AL is especially useful in cases where large portions of data are unlabeled and manual labeling is expensive [3]. Selecting the best single annotated frame to train OSVOS represents a particularly hard problem in AL, starting to learn with no initial labeled instances, i.e., the cold start problem [26].

Within AL, we are particularly interested in error reduction. Error reduction is an intuitive sub-field that directly optimizes the objective of interest and produces more accurate learners with fewer labeled instances than uncertainty or hypothesis-based AL approaches [34]. However, rather than going through all video frames and then formally predicting the expected error reduction associated with any one annotation frame, BubbleNets simplifies the problem by only comparing the relative performance of two frames at a time. By combining our decision framework with a bubble sort, we iterate this selection process across the entire video and promote the frame with the best relative performance to be our selected annotation frame.

Within computer vision, previous AL work includes measures to reduce costs associated with annotating images and selecting extra training frames *after* using an initial set of user annotations. Cost models predicting annotation times can be learned using a decision-theoretic approach [38, 40]. Other work has focused on increasing the effectiveness of crowd-sourced annotations [39]. To improve tracking performance, active structured prediction has been used to suggest extra training frames after using an initial set of user annotations [43]. Within VOS, other work in-

creases segmentation accuracy by having a user review segmentations and then add annotations on frames with poor performance [4]. The DAVIS 2018 challenge includes emphasis on maximizing segmentation performance with decreased user annotation time [5]. In contrast, we are not estimating annotation costs or selecting extra annotation frames. To support all semi-supervised VOS methods without increasing user effort, we are selecting a single frame for annotation that increases performance.

3. BubbleNets

We design an artificial neural network, BubbleNets (BN), that learns to suggest video frames for annotation that improve video object segmentation (VOS) performance. To learn performance-based frame selection on our custom network, we generate our own labeled training data. Labeled video data are expensive, so we design our network loss to learn from fewer initial frame labels, as discussed in Section 3.1. In Section 3.2, we introduce our deep bubble sorting framework that uses BN performance predictions to select a single frame for annotation. We provide details for our BN architecture in Section 3.3. In Section 3.4, we present our BN implementation for VOS with complete training and configuration details.

3.1. Predicting Relative Performance

Assume we are given a set of m training videos wherein each video has n frames with labels corresponding to some performance metric, $y \in \mathbb{R}$, which we leave unspecified here but define in Section 3.4.1. Our goal is to learn to select the frame with the greatest performance from each video.

One way to accomplish this task is to use the entire video as input to a network (e.g., using an LSTM or 3D-ConvNet [6]) and output the frame index with the greatest predicted performance; however, this approach only has m labeled training examples. A second way to formulate this problem is to use individual frames as input to a network and output the predicted performance of each frame. Using this formulation, the frame with the maximum predicted performance can be selected from each video and there are $m \times n$ labeled training examples. While this is a significant improvement over m examples, the second formulation only provides one training example per frame, which, for complicated and high annotation-cost processes like video object segmentation, makes the task of generating enough data to train a performance-prediction network impractical.

To that end, instead of directly estimating the predicted performance y of each training frame, BN predicts the *relative* difference in performance of two frames being compared (i.e., $y_i - y_j$ for frames i and j from the same video). This difference may seem trivial, but it effectively increases the number of labels and training examples from $m \times n$ to $m \times \binom{n}{2} \approx \frac{mn^2}{2}$.

To further increase the number of unique training examples and increase BN’s accuracy, we use k random video reference frames as an additional network input. When predicting the relative performance between two frames, additional consideration can be given to the frame that better represents the reference frames. Thus, similar to architectures that process entire videos, reference frames provide some context for the video as a whole. We find that reference frames not only increase BN’s accuracy in practice but also increase the number of training examples from $m \times \binom{n}{2}$ to $m \times \binom{n}{k+2} \approx \frac{mn^{(k+2)}}{k+2}$.

Finally, we define our performance loss function as:

$$\mathcal{L}(\mathbf{W}) := |(y_i - y_j) - f(x_i, x_j, X_{\text{ref}}, \mathbf{W})|, \quad (1)$$

where \mathbf{W} are the trainable parameters of BN, y_i is the performance label associated with the i th video frame, x_i is the image and normalized frame index associated with the i th video frame, X_{ref} is the set of k reference images and frame indices, and f is the predicted relative performance. For later use, denote the normalized frame index for the i th frame of an n -frame video as

$$I_i = \frac{i}{n}. \quad (2)$$

Including I as an input enables BN to also consider temporal proximity of frames for predicting performance.

3.2. Deep Bubble Sorting

Assume we train BubbleNets to predict the relative performance difference of two frames using the loss function (1) from Section 3.1. To select the frame with the greatest performance from a video, we use BN’s relative performance predictions within a deep bubble sorting framework, iteratively comparing and swapping adjacent frames until we identify the frame with the greatest predicted relative (and overall) performance.

Our deep bubble sorting framework begins by comparing the first two video frames. If BN predicts that the preceding frame has greater relative performance, the order of the two frames is swapped. Next, the leading frame is compared (and potentially swapped) with the next adjacent frame, and this process passes forward until reaching the end of the video (see Figure 2). The frame ranked highest at the end of the sort is selected as the predicted best-performing frame.

Normally, bubble sort is deterministic and only needs one pass through a list to promote the greatest element to the top; conversely, our deep bubble sorting framework is stochastic. BN uses k random video reference frames as input for each prediction, and using a different set of reference frames can change that prediction; thus, a BN comparison for the same two frames can change. While bubble sort’s redundancy is sub-optimal relative to other comparison sorts in many applications [20], revisiting previous comparisons

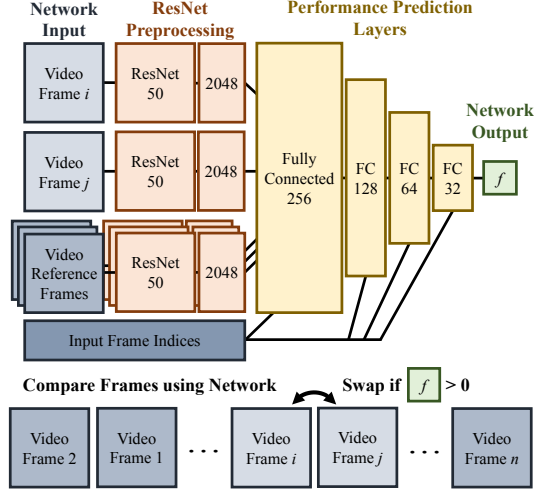


Figure 2. **BubbleNets Framework:** Deep sorting compares and swaps adjacent frames using their predicted relative performance.

is particularly effective given BN’s stochastic nature. Accordingly, our deep bubble sorting framework makes n forward passes for an n -frame video, which is sufficient for a complete frame sort and increases the likelihood that the best-performing frame is promoted to the top.

One way to increase BN’s consistency is to batch each network prediction over multiple sets of video reference frames. By summing the predicted relative performance over the entire batch, we reduce the variability of each frame comparison. However, two consequences of increasing batch size are: 1) increasing the chance of hitting a local minimum (i.e., some frame pairs are ordered incorrectly but never change) and 2) increasing execution time. In Section 4, we perform an ablation study to determine the best batch size for our specific application.

Although BN is not explicitly trained to find the best-performing frame in a video, our complete deep bubble sorting framework is able to accomplish this task, as shown in Figure 3. Even in cases where the best performing frames are not promoted to the top, an important secondary effect of our deep sorting framework is demoting frames that lead to poorer performance (e.g., Frame 20 in Figure 1); avoiding such frames is critical for annotation frame selection in video object segmentation.

3.3. BubbleNets Architecture

Our BubbleNets architecture is shown in Figure 2. The input has two comparison images, three reference images, and normalized indices (2) for all five frames. Increasing the number of reference frames, k , increases video-wide awareness for predicting relative frame performance but also increases network complexity; in practice, we find that $k = 3$ is a good compromise. The input images are processed using a base Residual Neural Network

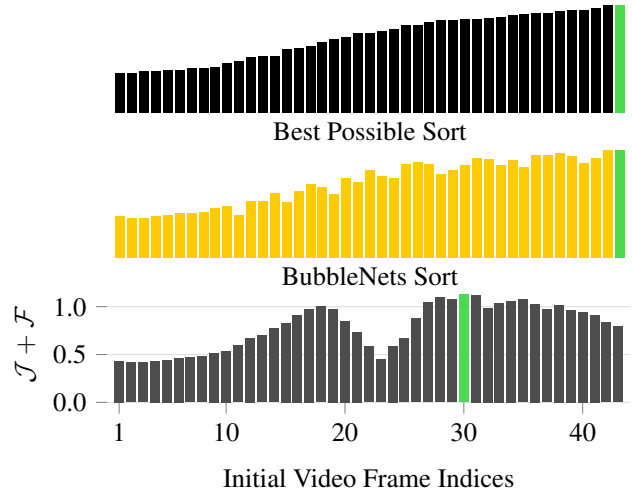


Figure 3. **BubbleNets Prediction Sort of Motorbike Video.** The green bar is the annotated training frame selected by BubbleNets.

(ResNet 50, [15]) that is pre-trained on ImageNet, which has been shown to be a good initialization for segmentation [9] and other video tasks [47]. Frame indices and ResNet features are fed into BN’s performance prediction layers, which consist of four fully-connected layers with decreasing numbers of neurons per layer. All performance prediction layers include the normalized frame indices as input and use a Leaky ReLU activation function [24]; the later three prediction layers have 20% dropout for all inputs during training [36]. After the performance prediction layers, our BN architecture ends with one last fully connected neuron that is the output relative performance prediction $f(x_i, x_j, X_{\text{ref}}, \mathbf{W}) \in \mathbb{R}$ in (1).

3.4. BubbleNets Implementation for Video Object Segmentation

Assume a user wants to segment an object in video and provides an annotation of that object in a single frame. Because annotating video data is time consuming, we use BubbleNets and deep sorting to automatically select the annotation frame for the user that results in the best segmentation performance possible. We segment objects from the annotated frame in the remainder of the video using One-Shot Video Object Segmentation (OSVOS) [4].

3.4.1 Generating Performance Labels for Training

Generating performance-based labels to train BN requires a quantitative measure of performance that is measurable on any given video frame. For our VOS performance measure, we choose a combination of region similarity \mathcal{J} and contour accuracy \mathcal{F} . Region similarity (also known as intersection over union or Jaccard index [11]) provides an intuitive, scale-invariant evaluation for the number of misla-

beled foreground pixels with respect to a ground truth annotation. Given a foreground mask M and ground truth annotation G , $\mathcal{J} = \frac{M \cap G}{M \cup G}$. Contour accuracy evaluates the boundary of a segmentation by measuring differences between the closed set of contours for M and G [31]; \mathcal{F} is also correlated with \mathcal{J} [14, Figure 5]. Using \mathcal{J} and \mathcal{F} , we define a frame performance label for loss function (1) as

$$y_i := \frac{1}{n} \sum_{j=1}^n \mathcal{J}_j + \mathcal{F}_j, \quad (3)$$

where y_i is i th label of an n -frame video and $\mathcal{J}_j + \mathcal{F}_j$ is the performance on frame j after using frame i for annotation. In simple words, y_i is the video-wide mean performance that results from selecting the i th frame for annotation.

We use our performance label (3) to generate BN training data. To avoid labeling costs for annotating BN-selected frames and evaluating segmentation performance, we use a previously annotated VOS dataset. Our ideal dataset contains many examples and is fully-annotated to provide BN the complete set of video frames for annotation selection. We give full consideration to the datasets listed in Table 1 [22, 31, 33, 37, 45]. YouTube-VOS contains the most annotated frames, but the validation set provides annotations on only the first video frame and the training set provides annotations only on every fifth frame. SegTrackv2 has the most annotated frames per video, but this metric is skewed by a handful of long videos and the majority of SegTrackv2 videos contain 40 frames or fewer (see Figure 4). Accordingly, we use the DAVIS 2017 training set, which contains the most examples of the fully annotated datasets.

Using the DAVIS 2017 training set, we train OSVOS for 500 iterations on every frame and find the resulting performance label (3). For videos with multiple annotated objects, performance labels are generated for each object on every frame. Preprocessing the dataset takes about a week on a dual-GPU (GTX 1080 Ti) machine but has many benefits. First, BN can train without running OSVOS, which significantly decreases training time. Second, we know the ground truth performance of every frame, so we can evaluate the overall deep sorting framework (e.g., seeing which frames are under- or over-promoted in Figure 3). Finally, we can compare performance against several simple frame selection strategies and know the best and worst frame selections possible for each video in the dataset.

3.4.2 Five BubbleNets Configurations and Training

To test the efficacy of new concepts and establish best practices, we implement five BN configurations for VOS. The first configuration (BN_0) uses the standard BN architecture in Section 3.3. The second and third configurations are similar to BN_0 but use No Input Frame Indices (BN_{NFI}) or No

Table 1. **Dataset Metrics.** Most of the SegTrackv2 videos and all of the YT-VOS videos have less than 40 annotated frames.

	DAVIS 2017		DAVIS	SegTrack	YT-VOS
Number of	Train	Val.	'16 Val.	v2	(1st 1,000)
Objects	144	61	20	24	1,000
Videos	60	30	20	14	607
Annotated Frames	4,209	1,999	1,376	1,066	16,715
Object Annotations	10,238	3,984	1,376	1,515	26,742
Annotated Frames Per Video					
Mean	70.2	66.6	68.8	76.1	27.5
Median	71	67.5	67.5	39	30
Range	25–100	34–104	40–104	21–279	8–36
Coef. of Variation	0.22	0.31	0.32	1.03	0.29

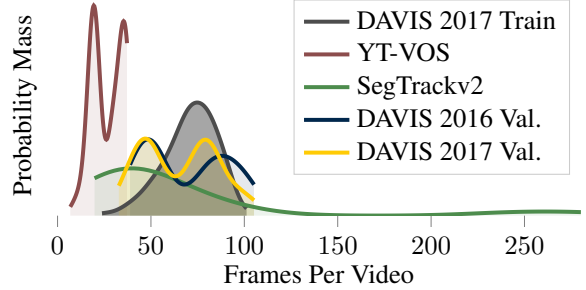


Figure 4. PMF for Annotated Frames Available Per Video.

Reference Frames (BN_{NRF}). The fourth and fifth configurations are similar to BN_0 but use loss functions modified from \mathcal{L} (1) to predict Single-frame Performance (BN_{LSP}) or bias toward middle Frame selection (BN_{LFF}).

BN_{LSP} 's single-frame performance loss is defined as:

$$\mathcal{L}_{\text{SP}}(\mathbf{W}) := |y_i - f(x_i, x_{\text{ref.}}, \mathbf{W})|, \quad (4)$$

where y_i is the single performance label for frame i . Alternatively, BN_{LFF} 's middle-frame biased loss is defined as:

$$\mathcal{L}_{\text{F}}(\mathbf{W}) := |(y_i - y_j) - (d_i - d_j) - f(x_i, x_j, x_{\text{ref.}}, \mathbf{W})|, \quad (5)$$

where d_i is the distance between frame i and the middle frame. Using the normalized index from (2), we find d_i as:

$$d_i = \lambda |I_i - I_{\text{MF}}|, \quad (6)$$

where $I_{\text{MF}} = 0.5$ is the normalized middle frame index and $\lambda = 0.5$ determines the relative emphasis of middle frame bias in (5). The intuition behind (5) is simple. In addition to predicting the performance difference between frames i and j , BN_{LFF} will learn to consider distance of each frame from the middle of the video. Given no predicted performance difference, the network will simply fall back on the frame closest to the middle, which is shown in Section 4 to be an effective annotation choice. To help BN_{LFF} learn the additional frame-based loss, we remove all network layer dropout associated with the frame input indices.

All five configurations are trained using the labeled DAVIS 2017 training data described in Section 3.4.1. To

Table 2. BubbleNets Configurations.

Config. ID	Input		Loss Function	Total Training		DAVIS 2017	
	Frame Indices	Ref. Frames		Iterations	Time	Val. \mathcal{J}	Mean \mathcal{F}
BN ₀	Yes	Yes	\mathcal{L} (1)	3,125	5m 11s	59.7	65.5
BN _{NIFI}	No	Yes	\mathcal{L} (1)	2,500	3m 52s	58.7	65.0
BN _{LF}	Yes	Yes	\mathcal{L}_F (5)	8,125	15m 30s	57.8	63.8
BN _{NRF}	Yes	No	\mathcal{L} (1)	3,125	2m 20s	55.4	62.3
BN _{LSP}	Yes	Yes	\mathcal{L}_{SP} (4)	1,875	2m 32s	55.1	62.3

decrease training time, all DAVIS 2017 training frames are preprocessed through the ResNet portion of the architecture, which does not change during BN training. We use a batch size of 1,024 randomly selected videos; each video uses up to five frames that are randomly selected without replacement (e.g., two comparison and three reference). We add an L1 weight regularization loss with a coefficient of 2×10^{-6} , and use the Adam Optimizer [19] with a 1×10^{-3} learning rate. The number of training iterations and training time for each configuration is summarized in Table 2.

We evaluate all models using the original bubble sorting framework, although BN_{LSP} requires two forward network passes per sort comparison and BN_{NRF} is deterministic without the random reference frames. Tasked with learning frame-based loss and frame performance differences, BN_{LF} requires the most training iterations of all BN networks. BN_{LSP} trains in fewer iterations due to simplified loss, and both BN_{LSP} and BN_{NRF} train faster due to fewer input images. As shown in Table 2, the BN₀ model outperforms BN_{LSP} and BN_{NRF}, justifying our claims in Section 3.1 for using relative frame performance and reference frames.

4. Experimental Results

4.1. Setup

Our primary experiments and analysis use the DAVIS 2017 Validation set. As with the training set in Section 3.4.1, we find the segmentation performance for every possible annotated frame, which enables us to do a complete analysis that includes the best and worst possible frame selections and simple frame selection strategies. We determine the effectiveness of each frame selection strategy by calculating the mean $\mathcal{J} + \mathcal{F}$ for the resulting segmentations on the entire dataset; the mean is calculated on a per video-object basis (e.g., a video with two annotated objects will contribute to the mean twice). Best and worst frame selections are determined using the combined $\mathcal{J} + \mathcal{F}$ score for each video object. The simple frame selection strategies are selecting the first frame (current VOS standard), middle frame (found using floor division of video length), last frame, and a random frame from each video for each object. Finally, because BN results can vary from using random reference frames as input, we only use results from the first run of each configuration (same with random frame selection).

Table 3. Ablation Study on DAVIS 2017 Val. Set: Study of BN input batch size for bubble sort comparisons and end performance.

Batch Size	Performance ($\mathcal{J} + \mathcal{F}$)			Mean Video Sort Time
	BN ₀	BN _{NIFI}	BN _{LF}	
1	124.1	122.9	120.5	3.88 s
3	125.2	122.0	121.6	4.83 s
5	125.2	123.8	121.7	5.32 s
10	125.2	122.0	120.3	6.52 s
20	123.6	123.4	120.7	9.34 s

4.2. Ablation Study

We perform an ablation study to determine the best batch size for BN predictions. Recall from Section 3.2 that batches reduce variability by using multiple sets of random reference frames. As shown in Table 3, a batch size of 5 leads to the best performance for all BN configurations and is chosen as the standard setting for all remaining results. The mean video sort times in Table 3 are for BN_{LF}, which consistently has the highest sort times. As a practical consideration, we emphasize that the frame selection times in Table 3 are negligible compared to the time it takes a user to annotate a frame [5].

4.3. DAVIS Validation

Complete annotated frame selection results for the DAVIS 2016 and 2017 validation sets are provided in Table 4. To put these results in perspective, the current difference in $\mathcal{J} + \mathcal{F}$ for the two leading VOS methods on the DAVIS 2016 Val. benchmark is 2.1 [25, 42].

For first frame selection, it is worth acknowledging that both datasets intend for annotation to take place on the first frame, which guarantees that objects are visible for annotation (in some videos, objects become occluded or leave the view). Despite this advantage, middle frame selection outperforms first frame selection on both datasets overall and on 3/5 of the videos on DAVIS 2017 Val. In fact, on both datasets first frame selection is, on average, closer to the worst possible frame selection than the best. Last frame selection has the worst performance and, using the coefficient of variation, the most variable relative performance. Finally, the best performing annotation frame is never the first or last frame for any DAVIS validation video.

Middle frame selection has the best performance of all simple strategies. We believe that the intuition for this is simple. Because the middle frame has the least cumulative temporal distance from all other frames, it is on average more representative of the other frames with respect to annotated object positions and poses. Thus, the middle frame is, on average, the best performing frame for segmentation.

All BN configurations outperform the simple selection strategies, and BN₀ performs best of all BN configurations. When selecting different frames, BN₀ beats middle frame selection on 3/5 videos and first frame selection on 4/5 videos for DAVIS 2017 Val. By comparing the performance

Table 4. Dataset Annotated Frame Selection Results.

Annotation Frame Selection	Segmentation Performance ($\mathcal{J} + \mathcal{F}$)			
	Mean	Median	Range	Coef. of Variation
DAVIS 2017 Val.				
Best	141.2	143.2	14.9–194.9	0.26
BN ₀	125.2	128.9	7.6–194.2	0.34
BN _{NIFI}	123.8	129.9	8.7–194.2	0.35
BN _{CF}	121.7	128.0	7.6–194.3	0.38
Middle	119.2	124.0	7.6–193.6	0.41
Random	116.5	119.7	1.6–193.2	0.38
First	113.3	117.2	3.5–192.5	0.39
Last	104.7	110.3	4.4–190.1	0.42
Worst	86.3	88.2	1.6–188.9	0.56
DAVIS 2016 Val.				
Best	171.2	176.3	130.6–194.9	0.11
BN ₀	159.8	168.5	72.6–194.5	0.18
BN _{NIFI}	157.3	165.7	72.6–194.5	0.18
BN _{CF}	155.6	170.5	72.6–193.8	0.21
Middle	155.2	169.5	77.1–193.8	0.21
First	152.8	153.4	115.2–191.7	0.15
Random	147.5	157.3	83.1–194.5	0.25
Last	147.5	153.0	72.0–189.6	0.23
Worst	127.7	141.3	68.3–188.9	0.31

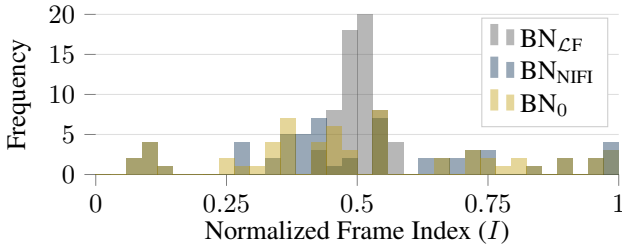


Figure 5. Frame-Selection Locations in Video: Normalized indices (2) of all BN annotation frame selections on DAVIS ‘17 Val.

of BN₀ and BN_{NIFI}, we find that BN₀’s use of normalized frame indices (2) is beneficial for performance.

Finally, it is clear from the frame-selection locations in Figure 5 that BN_{CF}’s modified loss function (5) successfully biases selections toward the middle of each video.

4.4. Results on Datasets with Limited Frames

Annotated frame selection results for SegTrackv2 and YouTube-VOS are provided in Table 5. As emphasized in Section 3.4.1, the videos in these datasets have a limited number of frames available for annotation, which limits the effectiveness of BN frame selection. Because the YouTube-VOS validation set only provides annotations on the first frame, we instead evaluate on the first 1,000 objects of the YouTube-VOS training set, which provides annotations on every fifth frame. This reduces the number of candidate annotation frames that BN can compare, sort, and select to one fifth of that available in a standard application for the same videos. While all BN configurations outperform first and last frame selection, BN_{CF} is the only configuration that consistently outperforms all other selection strategies. We postulate that the additional bias of BN_{CF} toward

Table 5. Results on Datasets with Limited Frames Per Video.

Annotation Frame Selection	Segmentation Performance ($\mathcal{J} + \mathcal{F}$)			
	Mean	Median	Range	Coef. of Variation
SegTrackv2				
BN_{CF}	134.7	145.9	14.3–184.6	0.32
Middle	134.5	143.5	14.3–182.8	0.32
BN _{NIFI}	134.3	144.2	33.9–178.5	0.30
BN ₀	130.6	127.3	50.0–183.2	0.30
Last	123.6	130.4	14.3–178.4	0.36
First	122.3	122.5	45.8–181.7	0.31
YT-VOS (1st 1,000)				
BN_{CF}	115.5	126.6	0.0–197.3	0.46
Middle	115.0	124.2	0.0–196.2	0.46
BN _{NIFI}	111.8	121.0	0.0–196.3	0.47
BN ₀	110.4	121.5	0.0–194.1	0.49
First	107.3	114.0	0.0–196.3	0.49
Last	101.2	108.1	0.0–195.4	0.56

Table 6. Cross Evaluation of Benchmark Methods: OSVOS and OnAVOS DAVIS ‘17 Val. results using identical frame selections.

Segmentation Method	Frame Selection and DAVIS $\mathcal{J} + \mathcal{F}$ Mean				
	First	Middle	BN _{CF}	BN _{NIFI}	BN ₀
OSVOS	56.6	59.6	60.8	61.9	62.6
OnAVOS	63.9	68.4	68.5	68.4	69.2

Table 7. Frames Per Video and Relative Performance: BN performances relative to first frame on DAVIS 2017 Validation.

Videos from DAVIS 2017 Val.	Number of Frames	Relative Mean ($\mathcal{J} + \mathcal{F}$)		
		BN ₀	BN _{NIFI}	BN _{CF}
10 Longest	81–104	+ 11.8%	+ 10.9%	+ 4.0%
All	34–104	+ 10.5%	+ 9.3%	+ 7.4%
10 Shortest	34–43	+ 4.9%	+ 5.0%	+ 3.3%

Figure 6. BN_{CF}–Middle Frame Comparison: Two best (left) and worst (right) BN_{CF} selections relative to the middle frame.

index-based selections made this configuration more robust to reductions in candidate annotation frames.

4.5. Results on Different Segmentation Methods

Cross evaluation results for different segmentation methods are provided in Table 6. All BN configurations select annotation frames that improve the performance of OnAVOS, despite BN training exclusively on OSVOS-generated labels. Nonetheless, the label-generating formulation in Section 3.4.1 is general to other semi-supervised VOS methods; thus, new BN training labels can always be generated for other methods. Note that first frame results in Table 6 differ from the online benchmark due to dataset-specific configurations (e.g., [41]), non-deterministic components, and our segmenting and evaluating objects from multi-object videos separately, which is more challenging.

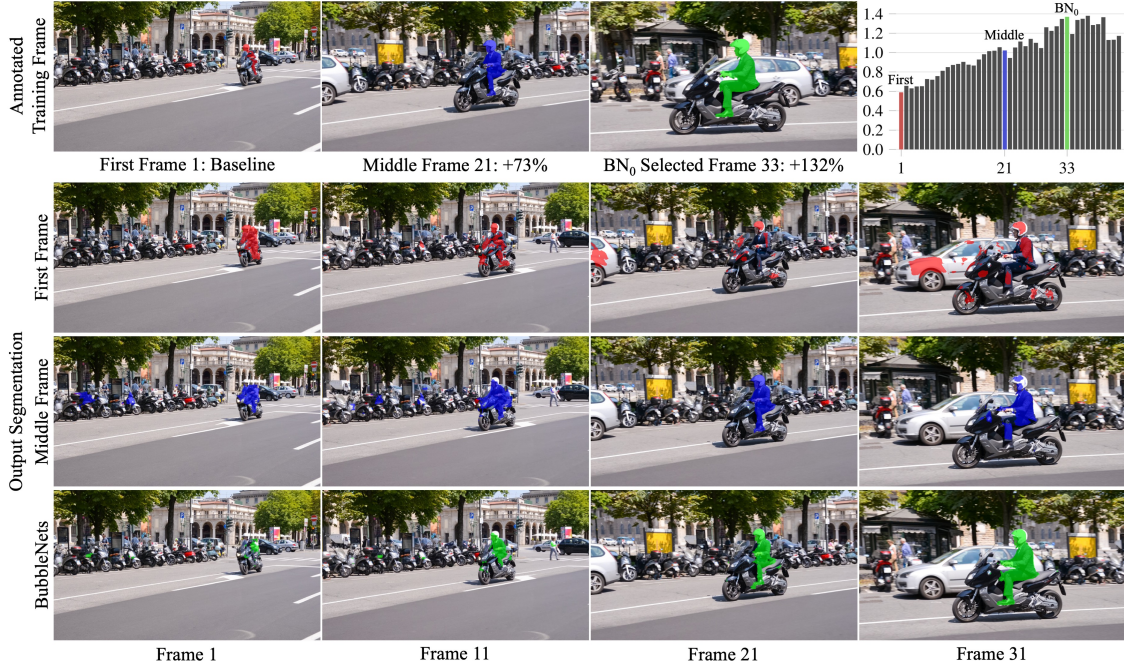


Figure 7. **Qualitative Comparison on DAVIS 2017 Validation Set:** Segmentations from different annotated frame selection strategies.

4.6. Final Considerations for Implementation

Selecting the middle frame for annotation is the best performing simple selection strategy on all datasets and is easy to implement in practice. However, BN_{LF} is more reliable than middle frame selection and results in better segmentation performance on all datasets. As shown in Figure 5, BN_{LF} selects frames close to the middle of each video, but deviates toward frames that, on average, result in better performance than the middle frame (see Tables 4 & 5). On DAVIS 2017 Val., BN_{LF} deviations from the middle frame results in better performance 70% of the time. We believe the underlying mechanism for this improvement is recognizing when the middle frame exhibits less distinguishable ResNet features or is less representative of the video reference frames. To demonstrate beneficial and counterproductive examples of this behavior, the two best and worst BN_{LF} selections relative to the middle frame on DAVIS 2017 Val. are shown with relative performance % in Figure 6.

BN_0 has the greatest relative segmentation improvements over simple selection strategies on the DAVIS validation datasets (see example comparison in Figure 7). However, this performance did not translate to datasets with a limited number of annotation frames available. To determine if this is due to domain shift of fewer frames, we analyze the 10 longest and shortest videos from DAVIS 2017 Val. in Table 7 as an additional experiment. The key result is that BN_0 and BN_{NFI} 's relative performance gains double once approximately forty annotation frames are available. This is encouraging as most real-world videos have many

more frames available for annotation, which is conducive for BN_0 's best annotated frame selection results.

5. Conclusions

We emphasize that automatic selection of the best-performing annotation frames for video object segmentation is a *hard* problem. Still, as video object segmentation methods become more learning-based and data-driven, it is critical that we make the most of training data and users' time for annotation. The most recent DAVIS challenge has shifted focus toward improving performance given limited annotation feedback [5]. However, we demonstrate in this work that there are already simple strategies available that offer a significant performance improvement over first frame annotations without increasing user effort; likewise, our BubbleNets framework further improves performance using learned annotated frame selection. To continue progress in this direction and improve video object segmentation algorithms in practice, dataset annotators should give full consideration to alternative frame selection strategies when preparing future challenges.

Finally, while the current BubbleNets implementation is specific to video object segmentation, it is more widely applicable. In future work, we plan to apply BubbleNets to improve performance in other video-based applications.

Acknowledgements

This work was partially supported by the DARPA MediFor program under contract FA8750-16-C-0168.

References

- [1] S. Avinash Ramakanth and R. Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2
- [2] L. Bao, B. Wu, and W. Liu. CNN in MRF: video object segmentation via inference in A cnn-based higher-order spatio-temporal MRF. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2
- [3] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, and M. Sedlmair. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):298–308, Jan 2018. 2
- [4] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3, 4
- [5] S. Caelles, A. Montes, K. Maninis, Y. Chen, L. V. Gool, F. Perazzi, and J. Pont-Tuset. The 2018 DAVIS challenge on video object segmentation. *CoRR*, abs/1803.00557, 2018. 1, 2, 3, 6, 8
- [6] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3
- [7] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1
- [8] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1
- [9] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 4
- [10] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2
- [11] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 4
- [12] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *British Machine Vision Conference (BMVC)*, 2014. 1
- [13] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. Jumpcut: non-successive mask transfer and interpolation for video cutout. *ACM Trans. Graph.*, 34(6):195, 2015. 2
- [14] B. A. Griffin and J. J. Corso. Tukey-inspired video object segmentation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 5
- [15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4
- [16] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2
- [17] W. D. Jang and C. S. Kim. Online video object segmentation via convolutional trident network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7474–7483, July 2017. 1
- [18] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for object tracking. *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2017. 2
- [19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014. 6
- [20] D. Knuth. *The Art of Computer Programming*, volume 1–3. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998. 3
- [21] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. 1
- [22] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *The IEEE International Conference on Computer Vision (ICCV)*. 1, 2, 5
- [23] X. Li, Y. Qi, Z. Wang, K. Chen, Z. Liu, J. Shi, P. Luo, C. C. Loy, and X. Tang. Video object segmentation with re-identification. *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2017. 2
- [24] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013. 4
- [25] K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. V. Gool. Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. 1, 2, 6
- [26] A. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *International Conference on Machine Learning (ICML)*, 1998. 2
- [27] N. Mrki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 743–751, June 2016. 2
- [28] S. W. Oh, J.-Y. Lee, K. Sunkavalli, and S. J. Kim. Fast video object segmentation by reference-guided mask propagation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2
- [29] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 1
- [30] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

- [31] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 5
- [32] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2
- [33] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 1, 2, 5
- [34] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012. 2
- [35] J. Shin Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1
- [36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. 4
- [37] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg. Motion coherent tracking using multi-label mrf optimization. *International journal of computer vision*, 100(2):190–202, 2012. 1, 2, 5
- [38] S. Vijayanarasimhan and K. Grauman. What’s it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1, 2
- [39] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision*, 108(1):97–114, May 2014. 2
- [40] S. Vijayanarasimhan, P. Jain, and K. Grauman. Far-sighted active learning on a budget for image and video recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2
- [41] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2017. 7
- [42] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *British Machine Vision Conference (BMVC)*, 2017. 2, 6
- [43] C. Vondrick and D. Ramanan. Video annotation and tracking with active learning. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 28–36. Curran Associates, Inc., 2011. 2
- [44] S. Wehrwein and R. Szeliski. Video segmentation with background motion models. In *British Machine Vision Conference (BMVC)*, 2017. 1
- [45] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 1, 2, 5
- [46] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2
- [47] L. Zhou, C. Xu, and J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, 2018. 4