

# Curriculum Learning for Recurrent Video Object Segmentation

Maria Gonzalez-i-Calabuig<sup>1</sup>, Carles Ventura<sup>2</sup>, and Xavier Gir-i-Nieto<sup>1</sup>

<sup>1</sup> Universitat Politcnica de Catalunya

{[maria.gonzalez.calabuig](mailto:maria.gonzalez.calabuig@upc.edu),[xavier.giro](mailto:xavier.giro@upc.edu)}@upc.edu

<sup>2</sup> Universitat Oberta de Catalunya [cventuraroy@uoc.edu](mailto:cventuraroy@uoc.edu)

**Abstract.** Video object segmentation can be understood as a sequence-to-sequence task that can benefit from the curriculum learning strategies for better and faster training of deep neural networks. This work explores different schedule sampling and frame skipping variations to significantly improve the performance of a recurrent architecture. Our results on the car class of the KITTI-MOTS challenge indicate that, surprisingly, an inverse schedule sampling is a better option than a classic forward one. Also, that a progressive skipping of frames during training is beneficial, but only when training with the ground truth masks instead of the predicted ones. Source code and trained models are available at <http://imatge-upc.github.io/rvos-mots/>.

**Keywords:** Video Object Segmentation, Recurrent Neural Networks, Curriculum Learning

## 1 Introduction

The optimization process of deep neural networks is greatly influenced by how training data is used. Curriculum learning [3] is a training strategy for machine learning that consists of presenting simple concepts to the model first to, gradually, increasing their complexity.

Our work proposes two training curriculums for a Recurrent Video Object Segmentation engine (RVOS) [8], a neural model for one-shot (or semi-supervised) video object segmentation (VOS). In this task, a binary mask of an object is provided for a single frame and the goal is predicting the mask of the selected object across the rest of the frames in the video sequence. RVOS architecture is based on an end-to-end recurrent Conv-LSTM [13] decoder that tracks objects across frames, with no need of any post-processing. The recurrent architecture makes RVOS a fast solution for the task, capable of processing more than 20 frames per second [1]. RVOS was originally tested on the DAVIS and YouTube-VOS datasets for one-shot video object segmentation. We show how RVOS struggles with the *cars* in the KITTI-MOTS dataset [9], whose videos are more crowded and challenging than DAVIS or YouTube-VOS. We improve the off-the-shelf RVOS baseline by modifying its training curriculum in two ways. First, with a schedule sampling [2] totally contrary to the one original one in

RVOS and, secondly, by gradually increasing the complexity of the task by sub-sampling video frames at training time.

## 2 Related Work

**Schedule Sampling** [2] offers an alternative to *teacher forcing* [10] where, during training time, the model has access to the ground truth label of the previous time-step in each new prediction. During inference, the model uses its predictions as input in the next training step. This may lead to exposure bias because of the discrepancy between training and inference and result in poor model performance. Schedule sampling takes benefit from teacher forcing while avoiding exposure bias by gradually replacing the ground-truth tokens by the models predictions. Three different decay schedules were proposed by Bengio et al. [2]: exponential, inverse sigmoid and linear. While Ren and Zemel [7] and Xu et al. [14] used a linear schedule in their training, Oh et al. [12] and RVOS [8] adopted a more drastic scheme, using ground truth labels in the first half of the training, and predicted masks in the second half. We have named this second approach as a *step* schedule, as in the well-known Heaviside step function.

**Frame Skipping** is a training curriculum in which video sequences are progressively sub-sampled in time so that the model is exposed to sequences with faster changes, even if synthetically generated. The limited sizes of the mini-batches typically force training with short sequences which, in the case of video, may be highly redundant if considering consecutive frames.

Frame Skipping was introduced in the Space-Time Memory Networks (STM) [6], inspired by [15] and related to their own previous model [5]. STMs increase gradually the amount of skipped frames, from 0 to 25. Wu et al. [11] achieved relevant gains when processing video streams at a *fast* and a *slow* frame rates in two different pathways that merge at the deepest layer.

## 3 Experiments

We have explored different schedule sampling and frame skipping strategies with the RVOS model [8] evaluated on the *car* class in the validation partition of the KITTI-MOTS benchmark [9]. The task addressed is the one-shot (or semi-supervised) video object segmentation (VOS) task, where a mask of the object is provided to the model to estimate the masks in the rest of the frames in the video sequence. All models are trained during a fixed amount of 40 epochs.

We adopt the official metrics for the MOT Challenge [9] to obtain quantitative results: sMOTSA, MOTSP, Recall and Precision. In all cases, the higher the metric, the better. However, instead of averaging the metrics per pixel as in the public benchmark, we have averaged them by sequence. Otherwise, the results over longer sequences would dominate over the rest.

Two different strategies have been considered when allocating memory in the GPUs for training: whether we considered a lower spatial resolution (256x448 pixel) and longer clips of 5 frames, or a higher spatial resolution (287x950) and

**Table 1.** Schedule sampling variations of one-shot VOS on KITTI-MOTS *cars*. Best values are shown in **bold** and second best values in **blue**.

	Image resolution	Batch size	Length clip	sMOTSA	MOTSP	Recall	Precision
Teacher Forcing	256x448	4	5	-16.57	73.98	32.81	43.62
	287x950	2	3	<b>4.24</b>	77.00	45.84	<b>57.87</b>
Forward Step	256x448	4	5	-6.83	68.12	37.38	49.70
	287x950	2	3	-11.70	75.68	46.47	47.63
Forward Linear	256x448	4	5	-2.29	72.97	41.00	53.64
	287x950	2	3	-5.58	76.76	46.72	51.53
Inverse Step	256x448	4	5	-1.57	73.17	42.79	55.00
	287x950	2	3	<b>8.90</b>	<b>77.90</b>	42.86	<b>60.33</b>
Inverse Linear	256x448	4	5	-4.77	73.35	<b>48.60</b>	53.06
	287x950	2	3	2.48	<b>77.87</b>	<b>47.12</b>	57.07

shorter clips of 3 frames. While the 287x950 definition matches the aspect ratio of the KITTI-MOTS dataset [9], the 256x448 one corresponds to the aspect ratio of the YouTube-VOS dataset [14], for which RVOS was originally trained.

The KITTI-MOTS competition addresses a zero-shot challenge while our work has been focused on addressing a one-shot challenge. RVOS has demonstrated better performance with one-shot learning, which has been the motivation for choosing this approach. For this reason, the obtained results will not be compared with other state of the art works. Our objective is to explore the impact of the curriculum learning strategies on the performance of this model.

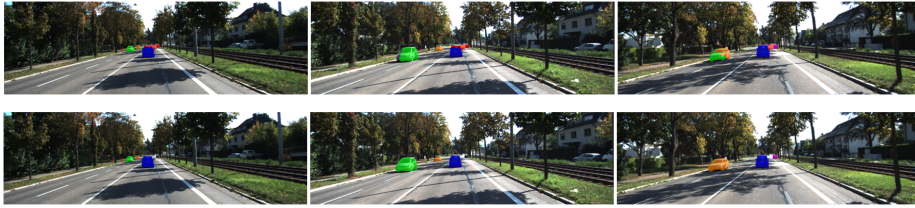
### 3.1 Schedule Sampling

Our experiments on schedule sampling consider the step and linear schedules in addition to the teacher forcing, provided as a baseline to compare with. The study extends to the non-conventional inverse variations for both the step and linear cases, inspired by the finding reported in [4]. The inverse variations actually defy the curriculum learning paradigm, as they start the training with the prediction of the model as references, and progress into a set up that considers only ground truth labels at the end.

The results presented in Table 1 indicate that actually the Forward Step curriculum adopted in the original RVOS baseline is the worst option, and that actually the best option is the inverse step approach. Figure 1 shows a fragment of a sequence in which the inverse step outperforms the baseline model.

### 3.2 Frame Skipping

Two frame skipping schemes were explored. In the *0 to 9* scheme, the number of skipped frames, which will be referred to as skipping step, is changed every 2 epochs. The total number of skipping steps is 10. The model starts training without skipping any frame and, gradually, increases the number of skipped



**Fig. 1.** Qualitative results on three non-consecutive frames comparing the baseline model (row 1) and the model with the best performance: inverse step (row 2). Compared to the inverse step strategy, during all the sequence, on the baseline model, a wrong mask in red is observed next to the blue instance. Also, the orange mask is confused by the green mask.

frames by 1 until 9 consecutive frames are skipped. The second scheme, the *1 to 5* one, halves the number of skipping steps from 10 to 5. In this case, the number of skipped frames is increased after 4 epochs, doubling the training time per skipping step.

These experiments are run with the RVOS baseline mode, which follows the Forward Step schedule sampling. On the first training phase, when using the ground-truth (GT) annotations, frame skipping is always used. During the second training phase, when the model’s predictions (Pred.) are used for training, we consider the two cases of skipping and non-skipping frames. We consider this hybrid approach because the difficulty of having to deal with the noisy predictions of the model may be overwhelming for our model when adding on top the temporal sub-sampling. During the second phase, when frame skipping is applied, the skipping step begins from 0 and increases to 9 again.

The results in Table 2 actually show that applying a frame skipping strategy during all training does not improve the performance of the model, maybe due to the difficulty of combining the two schemes. Instead, when using frame skipping only during the first training phase, the performance improves considerably for either set of experiments. As the sequences of KITTI-MOTS present a slow motion, the model benefits from training with this scheme. Analysing the results for both configurations, it can be seen how the best results are obtained with a frame skipping scheme of increasing from 1 to 5 skipped frames. The model benefits more when seeing changes but with enough time to process them.

### 3.3 Combination of Techniques

The previous experiments were performed as isolated experiments to fully understand the impact of each technique over the baseline model, the Forward Step. After obtaining these results, one extra experiment has been studied with the best configurations of each technique. The combination of inverse step schedule sampling and frame skipping gives an overall sMOTSA of 16.05, outperforming the results of 8.9 and -7.05 given by the inverse step schedule sampling and frame skipping from 1 to 5 respectively. This experiment has been tested with

**Table 2.** Frame skipping variations of one-shot VOS on KITTI-MOTS *cars*. Best values are shown in **bold** and second best values in **blue**.

	Image resolution	Batch size	Length clip	Skip @ GT	Skip @ Pred.	sMOTSA	MOTSP	Recall	Precision
No skip	256x448	4	5	No	No	-6,83	68,12	37,38	49,70
	287x950	2	3	No	No	-11,70	75,68	46,47	47,63
0 to 9	256x448	4	5	Yes	Yes	-39,39	58,30	1,57	3,33
	287x950	2	3	Yes	Yes	-17,66	74,99	46,70	50,00
	256x448	4	5	Yes	No	<b>-0,87</b>	74,73	<b>49,43</b>	<b>55,49</b>
	287x950	2	3	Yes	No	-8,18	<b>76,92</b>	44,67	48,21
1 to 5	256x448	4	5	Yes	Yes	-43,44	70,43	27,16	32,06
	287x950	2	3	Yes	Yes	-22,87	75,20	41,77	45,99
	256x448	4	5	Yes	No	<b>0,51</b>	<b>79,10</b>	39,26	53,57
	287x950	2	3	Yes	No	-7,05	75,86	<b>53,00</b>	<b>54,49</b>

the larger image resolution, as the performance on the inverse step with this configuration obtained the highest value among all the other experiments.

## 4 Conclusions

This work has shown how the curriculum learning greatly affects the performance of a deep neural network trained for the task of one-shot video object segmentation. The two techniques explored, schedule sampling and frame skipping, have brought significant gains to the RVOS model. These results encourage further research for a complete understanding and characterization of the techniques, especially in the surprising findings that an inverse step set up may result in better results. However, the low values of the quantitative results also invite to explore these curriculum learning with better performing architectures that may produce more stable and confident results. Future work includes exploring these strategies in other datasets as well as further research on the combination of the strategies with the best results.

## Acknowledgements

This work was partially supported by the Spanish Ministry of Economy and Competitivity and the European Regional Development Fund (ERDF) under contract TEC2016-75976-R and RTI2018-095232-B-C22. We gratefully acknowledge the support of NVIDIA Corporation with the donation of GPUs.

## References

1. Athar, A., Mahadevan, S., Ošep, A., Leal-Taixé, L., Leibe, B.: Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In: ECCV (2020)
2. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: NIPS (2015)
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: ICML (2009)
4. Huszr, F.: How (not) to train your generative model: Scheduled sampling, likelihood, adversary? (2015)
5. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Fast user-guided video object segmentation by interaction-and-propagation networks. In: CVPR (2019)
6. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (2019)
7. Ren, M., Zemel, R.S.: End-to-end instance segmentation with recurrent attention. In: CVPR (2017)
8. Ventura, C., Bellver, M., Girbau, A., Salvador, A., Marques, F., Giro-i Nieto, X.: Rvos: End-to-end recurrent network for video object segmentation. In: CVPR (2019)
9. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: Mots: Multi-object tracking and segmentation. In: CVPR (2019)
10. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* **1**(2), 270–280 (1989)
11. Wu, C.Y., Girshick, R., He, K., Feichtenhofer, C., Krahenbuhl, P.: A multigrid method for efficiently training video models. In: CVPR (2020)
12. Wug Oh, S., Lee, J.Y., Sunkavalli, K., Joo Kim, S.: Fast video object segmentation by reference-guided mask propagation. In: CVPR (2018)
13. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: NIPS (2015)
14. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: Youtube-vos: Sequence-to-sequence video object segmentation. In: ECCV (2018)
15. Yang, J., Reed, S.E., Yang, M.H., Lee, H.: Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In: NIPS (2015)