Improving Unsupervised Label Propagation for Pose Tracking and Video Object Segmentation*
Supplemental Material

Urs Waldmann[1,2][0000−0002−1626−9253], Jannik Bamberger[1], Ole Johannsen[1][0000−0002−7786−8516], Oliver Deussen[1,2][0000−0001−5803−2185], and Bastian Goldlücke[1,2][0000−0003−3427−4029]

[1] Department of Computer and Information Science, University of Konstanz, Germany
[2] Centre for the Advanced Study of Collective Behaviour, University of Konstanz, Germany
Corresponding author: `urs.waldmann@uni-konstanz.de`

Abstract. In the supplemental material, we first provide additional information including implementation details on our applications and case studies that we briefly introduced in Sec. 4 in our main paper. Then we report additional ablation studies and provide results. Next we report qualitative results for our method of joint tracking and keypoint propagation on our pigeons and JHMDB-test. We also report qualitative results for one-shot VOS on DAVIS2017val. Also, we report more quantitative results, i.e. for multi-object one-shot VOS on DAVIS2017val and SegTrackV2. Finally, we give details on our synthetic pigeon dataset and a rough expected inference speed for our applications and case studies. Please also check out our videos on https://urs-waldmann.github.io/improving-unsupervised-label-propagation/ for further insights into our pigeon keypoint tracking.

* This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: https://doi.org/10.1007/978-3-031-16788-1_15. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms.

Table of Contents

1 Additional Information on Applications and Case Studies

In this section of the supplemental material, we provide definitions together with implementation details of the encoders and decoders for the applications and case studies introduced briefly in Sec. 4 in our main paper.

1.1 Pose Tracking

Encoding keypoints requires some effort to transform the n points $K_{i,j} \in \mathbb{R}^2$, $j = 1, \dots, n$ for the ith frame into 2D functions. Commonly [13,6,12,21], each keypoint is encoded to a separate layer in the label function, thus $l = n$. The label encoding for the jth keypoint is then a 2D Gaussian with mean $K_{i,j} \odot [w/W, h/H]^T$, where $\odot$ denotes point-wise multiplication. Its standard deviation σ is a hyperparameter and should be chosen according to the feature size. Here, we found we can improve upon some existing implementations. As the label function has the resolution of the feature map, it is often only $\frac{1}{4}$ or even $\frac{1}{8}$ of the frame resolution. Therefore, some inaccuracies in the placement of the Gaussian function can occur if μ is not computed with sub-pixel precision. For example, UVC [13] scales the features to the feature size but then casts the floating point numbers to integers, thus discarding the sub-pixel precision. In contrast, we implement our encoding function in such a way that the peak of the Gaussian does not necessarily lie on an integral pixel position.

Decoding label functions to keypoints requires the approximate peak location for each label function. The intuitive solution of using the location of the maximum is unreliable, as the quality of the result hinges on a single point. Hence, some methods compute the mean or center of gravity of the top k positions to improve the robustness of the solution to outliers. Our method follows the latter approach of using the center of gravity not only for its improved robustness, but also because the results have sub-pixel precision. The keypoints are scaled up according to the frame size to feature size scale factor after decoding, therefore we need a sub-pixel precision at the feature level to achieve pixel-level accuracy at the frame scale. It is important to estimate the center of gravity from a small set of locations, because the majority of values in the label function are close to zero. Therefore, they only have very limited benefit to the localization of the peak but introduce relatively large errors compared to their contribution.

Implementation Details. For pose tracking of humans on JHMDB [7], we use the ResNet18 [4] feature extractor trained with UVC [13]. The training data is the same as in the original paper, i.e. MSCOCO [14] for the autoencoder and Kinetics [8] for the feature network. Feature vector normalization with the L_2 norm is applied to all feature maps. To improve the keypoint accuracy, we scale the 240×320 pixel source images of JHMDB up to 480×640 pixels. Each keypoint is encoded with a subpixel-accurate Gaussian function with $\sigma = 0.5$. Label decoding uses center of mass computed from the top 5 locations. Label propagation uses the 20 immediately preceding frames and the first frame as context frames. The affinity matrices are normalized with softmax per column and restricted to a square neighborhood with a side length of 25. The label propagation uses the top 20 reference locations over all reference frames at once.

1.2 Joint Tracking and Label Propagation

Tracking objects which occupy only a small part of the video frames, such as in our particular use case of video material of pigeons, poses two difficulties:

1. The affinity size becomes very large if the frame is processed at a high resolution.
2. The precision loss caused by processing at feature scale impedes the performance significantly, even if sub-pixel accuracy is considered during tracking.

We solve both of these problems simultaneously by tracking the objects with a fast single-object tracker. We choose UDT [19] for the tracking as it is trained in an unsupervised manner, has good performance and is very fast. The tracking allows us to crop a rectangle around the object of interest and resize it to a fixed size. The keypoint locations are transformed with the offset and scale of the resized cropping rectangle, such that we can perform label propagation only on the cropped region, see Fig. 1 in our main paper. This setup allows us to track the keypoints at a much higher resolution and simultaneously reduce the overhead of processing the remainder of the frame.

Implementation Details. We use UDT [19] as our object tracker. The feature network is trained on ImageNetVID [18]. We use scale estimation at three scales, an update factor of 0.01, regularization $\lambda = 0.1$ and a patch size of 125×125. For the remaining details on the tracker implementation, please refer to the original work [19]. We initialize the tracker with the bounding box containing all of the initial keypoints. We define the cropping region based on the tracked rectangle of size $h \times w$. The center of this rectangle defines the center of the cropping region. The cropping region is a square with side length $2 \cdot \max(h, w)$. If parts of this cropping region are outside of the image, we move the region to fit entirely within the image. This ensures, that the cropped image is not stretched or contains black regions. After the image is cropped, it is resized to 480×480 pixels with bicubic interpolation. Size is chosen to ensure that the object is large enough to achieve good accuracy while keeping the memory requirements for the label propagation under control.

1.3 Unsupervised Zero-Shot VOS

Mask Encoding and Decoding. We chose the number of label functions as $l = n + 1$ and encode each mask M_i^j as a separate layer by scaling it down to match the feature resolution of (h, w). The additional layer is the background layer. It is set to 1 at all locations where no other mask is placed. Decoding the masks is implemented as argmax_l, thus each pixel is occupied by the mask with the strongest response or the background. This is followed by bilinear mask upscaling up to resolution (H, W).

Unsupervised Mask Initialization: Pixel Location Selection per Attention Head. Pixels are selected to sum to a fixed percentage of the total mass. The selection is done in order of decreasing attention and results in a binary mask for each of the remaining attention heads. These binary masks are aggregated with a pixel-wise maximum to obtain a combined mask that should already be close to the object of interest. Since the mass-based selection does not take into account which pixels are contiguous we end up with some noise in the preliminary mask, which we reduce with a median filter.

Unsupervised Mask Initialization: CRF-based Mask Refinement. The mask prediction is slightly smoothed with a Gaussian function and normalized, such that the maximum values are slightly below 1.0. This prepared soft mask is then used as a prior for a CRF [10] to return a better fit to the input image. As a final step we apply another median filter to eliminate unwanted stray pixels from the mask.

Implementation Details. We chose the mass percentage as 0.5, and the quantile as 0.5. The median filter prior to mask refinement uses a kernel size of 3×3. To compute the prior of the mask refinement, we apply a Gaussian blur with $\sigma = 1.0$ to the preliminary mask and scale the values to fit in $[0.2, 0.8]$. Since the initial mask is at feature scale, it is scaled up with bilinear interpolation to match the input frame size. We perform 10 iterations of mask refinement. The final median filter uses a kernel size of 5×5. Compared to the baseline configuration for keypoint propagation we use only the first context frame for Z-VOS inference.

2 Additional Ablation Studies and Results

In this section of the supplemental material, we report more ablation studies and results (cf. Sec. 5.1 in our main paper).

2.1 Context Frames

In Fig. 1 we report qualitative results on the influence of the number of context frames for one-shot VOS (for quantitative results cf. Tab. 3). In Fig. 2 we show the influence of the number of context frames for our method of keypoint propagation with and without tracking (cf. Tabs. 1 and 2).

2.2 Pose Tracking

In Tabs. 1 and 2 we report results on ablation studies of various configuration elements for keypoint propagation without and with tracking respectively.

Our novel joint tracking and keypoint propagation pipeline (cf. Fig. 1 in our main paper) is the one that we use for pose tracking in Sec. 5.2 in our main paper since we achieve better results with that pipeline than with the core pipeline (cf. dashed box in Fig. 1 in our main paper). You see that by comparing Tabs. 1 and 2. For completeness, we still report ablation results for the core pipeline in Tab. 1.

2.3 One-Shot VOS

In Tab. 3 we report ablation results for one-shot VOS on DAVIS2017val. This is also the configuration that we use for unsupervised zero-shot VOS in Sec. 5.2 in our main paper. For the quantitative results in Tab. 4, we also report results where we change the number of context frames from 7 to 20. This change improves performance.

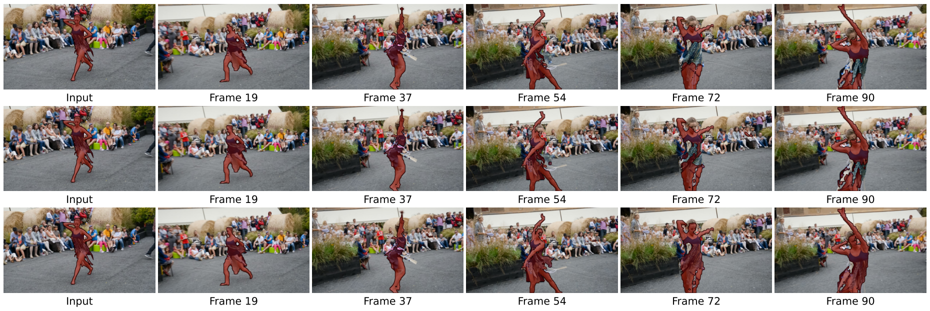


Fig. 1. *Qualitative comparison on the influence of the number of context frames for O-VOS.* Mask predictions for the *dance-twirl* video from DAVIS2017 with different numbers of context frames. From top to bottom the runs use 1, 7 and 20 context frames. Differences are especially visible in the later frames, e.g. frame 72 and frame 90.

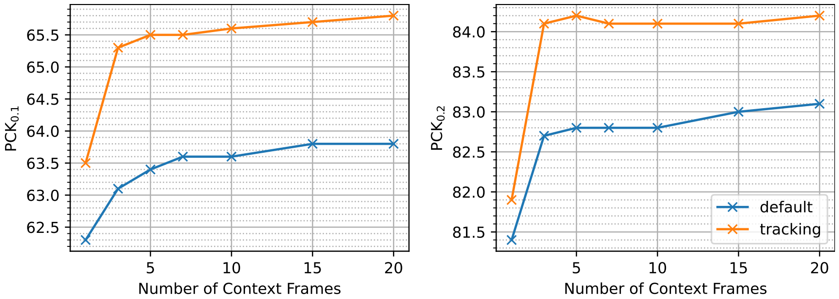


Fig. 2. *Influence of the number of context frames.* The two lines show the core pipeline configuration without tracking (cf. dashed box in Fig. 1 in our main paper) and the extended pipeline which also performs object tracking (cf. Fig. 1 in our main paper). The general observation is, that the $PCK_{0.1}$ increases with higher numbers of context frames, albeit with diminishing returns for large numbers of context frames. For $PCK_{0.2}$, the first few context frames bring noticeable performance improvements whereas later frames mostly help for the core pipeline.

Table 1. *Ablation: Keypoint propagation without tracking on the JHMDB test set.* The table shows the performance impact of various configuration elements. The baseline configuration is the core pipeline configuration for keypoint propagation without the tracking component (cf. dashed box in Fig. 1 in our main paper). The number of context frames is 7 in these experiments although we achieve better results with 20 context frames (cf. Fig. 2). The rows marked in grey show the result of the baseline configuration (repeated in each group for easier comparison). PCK is shown at thresholds $\tau = 0.1$ and $\tau = 0.2$. Δ_τ denotes the absolute difference to the baseline. mp: main paper, sm: supplemental material.

	$\mathrm{PCK}_{0.1}$	$\Delta_{0.1}$	$\mathrm{PCK}_{0.2}$	$\Delta_{0.2}$
Baseline	63.6	–	82.8	–
No Feature Normalization, mp Sec. 3.1	56.5	-7.1	78.5	-4.3
Batched Label Propagation, mp Sec. 3.2	61.8	-1.8	80.7	-2.1
No Subpixel-Accurate Labels, sm Sec. 1.1	63.0	-0.6	82.6	-0.3
Image Scale 320, mp Sec. 4.1	60.1	-3.4	79.7	-3.1
Affinity Top-K, mp Sec. 3.2				
1 (ArgMax)	55.9	-7.7	77.7	-5.1
5	62.1	-1.5	81.7	-1.1
10	63.2	-0.4	82.5	-0.3
13	63.3	-0.2	82.7	-0.2
15	63.4	-0.2	82.7	-0.1
17	63.5	-0.1	82.8	-0.0
20	63.6	–	82.8	–
23	63.5	-0.1	82.7	-0.1
25	63.5	-0.1	82.7	-0.2
Affinity Normalization, mp Sec. 3.2				
none	62.8	-0.7	82.4	-0.4
Softmax	40.3	-23.3	51.4	-31.4
UVC	62.8	-0.7	82.4	-0.4
UVC+Softmax	63.6	–	82.8	–
Local Affinity, Sec. mp 3.2				
3	62.4	-1.1	81.7	-1.1
5	63.3	-0.3	82.6	-0.2
12	63.6	–	82.8	–
Unrestricted	63.5	-0.1	82.6	-0.3
Label Standard Deviation, sm Sec. 1.1				
0.25	63.2	-0.3	82.7	-0.1
0.5	63.6	–	82.8	–
1.0	62.4	-1.2	80.2	-2.6
2.0	60.1	-3.4	77.5	-5.3

Table 2. *Ablation: Keypoint propagation with tracking on the JHMDB test set.* The table shows the performance impact of various configuration elements. The baseline configuration is our novel joint tracking and propagation configuration for keypoint propagation (cf. Fig. 1 in our main paper). The rows marked in grey show the result of the baseline configuration (repeated in each group for easier comparison). PCK is shown at thresholds $\tau = 0.1$ and $\tau = 0.2$. Δ_τ denotes the absolute difference to the baseline. mp: main paper, sm: supplemental material.

	PCK$_{0.1}$	$\Delta_{0.1}$	PCK$_{0.2}$	$\Delta_{0.2}$
Baseline	65.8	–	84.2	–
No Feature Normalization, mp Sec. 3.1	64.0	-1.8	82.5	-1.7
Batched Label Propagation, mp Sec. 3.2	61.6	-4.2	80.5	-3.7
No Subpixel-Accurate Labels, sm Sec. 1.1	65.5	-0.3	84.0	-0.2
Image Scale 320, mp Sec. 4.1	63.2	-2.6	82.1	-2.1
No Tracking, mp Sec. 4.1	63.9	-1.9	82.8	-1.4
Affinity Top-K, mp Sec. 3.2				
1 (ArgMax)	59.8	-6.0	80.8	-3.4
5	65.0	-0.8	83.8	-0.4
10	65.7	-0.0	84.3	0.1
13	65.7	-0.1	84.4	0.2
15	65.8	0.0	84.3	0.1
17	65.7	-0.1	84.4	0.2
20	65.8	–	84.2	–
23	65.7	-0.1	84.2	0.0
25	65.7	-0.1	84.0	-0.2
Affinity Normalization, mp Sec. 3.2				
none	65.3	-0.5	83.9	-0.3
Softmax	65.2	-0.6	83.8	-0.4
UVC	65.3	-0.5	83.9	-0.3
UVC+Softmax	65.8	–	84.2	–
Local Affinity, mp Sec. 3.2				
3	65.2	-0.6	83.5	-0.7
5	65.7	-0.1	84.0	-0.2
12	65.8	–	84.2	–
Unrestricted	65.6	-0.2	83.9	-0.3
Context Frames, mp Sec. 3.2				
1	63.5	-2.2	81.9	-2.3
3	65.3	-0.5	84.1	-0.1
5	65.5	-0.3	84.2	0.0
7	65.5	-0.3	84.1	-0.1
10	65.6	-0.2	84.1	-0.1
15	65.7	-0.1	84.1	-0.1
20	65.8	–	84.2	–
Label Standard Deviation, sm Sec. 1.1				
0.25	65.7	-0.1	84.1	-0.0
0.5	65.8	–	84.2	–
1.0	65.0	-0.8	82.4	-1.8
2.0	64.0	-1.7	80.8	-3.3

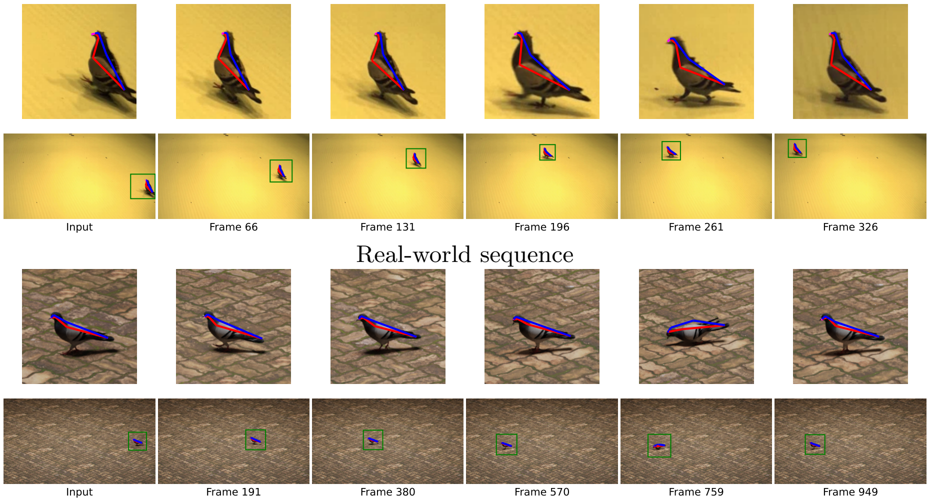
Table 3. *Ablation: O-VOS on DAVIS2017val.* The table shows the performance impact of various configuration elements. The rows marked in grey show the result of the baseline configuration (repeated in each group for easier comparison). This configuration is similar to the one used by DINO [2], with the only difference being the improved mask encoding. mp: main paper, sm: supplemental material.

	$\mathcal{J}\&\mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{J}_d$	$\mathcal{F}_m$	$\mathcal{F}_r$	$\mathcal{F}_d$
Baseline	71.79	68.40	81.42	15.23	75.18	86.89	18.69
No Feature Norm, mp Sec. 3.1	1.28	1.28	1.28	−2.22	1.28	1.28	−2.22
Batched Label Prop., mp Sec. 3.2	64.69	61.43	70.62	23.59	67.95	77.98	26.44
No Decoding Norm, sm Sec. 1.3	71.18	68.08	80.84	15.87	74.29	86.10	18.39
Affinity Top-K, mp Sec. 3.2							
3	71.69	68.33	81.56	15.57	75.05	86.88	19.66
5	71.79	68.40	81.42	15.23	75.18	86.89	18.69
10	71.61	68.25	80.88	14.46	74.98	85.56	17.37
15	71.04	67.66	79.77	14.45	74.42	84.67	16.91
20	70.38	67.06	79.27	14.54	73.70	84.24	17.08
25	70.02	66.72	78.67	14.55	73.31	84.21	16.86
Affinity Norm, mp Sec. 3.2							
Softmax	14.05	8.42	2.37	14.04	19.68	8.74	29.48
DINO and Softmax	63.09	59.81	70.99	23.78	66.38	78.33	28.30
None	71.68	68.29	81.28	14.72	75.07	86.80	17.81
DINO	71.79	68.40	81.42	15.23	75.18	86.89	18.69
Local Affinity, mp Sec 3.2							
3	66.73	63.64	75.07	22.56	69.81	81.59	26.19
5	69.87	66.40	78.36	19.92	73.35	85.53	22.20
7	71.40	68.04	80.61	15.99	74.76	87.10	19.54
12	71.79	68.40	81.42	15.23	75.18	86.89	18.69
Unrestricted	71.52	68.09	81.19	14.95	74.94	86.67	18.20
Context Frames, mp Sec. 3.2							
1	70.08	66.51	78.95	17.33	73.64	84.23	19.50
3	71.08	67.61	79.69	16.01	74.56	85.92	19.01
5	71.42	68.04	80.62	15.57	74.79	86.33	18.90
7	71.79	68.40	81.42	15.23	75.18	86.89	18.69
10	72.03	68.67	81.84	15.05	75.38	86.91	18.66
15	72.23	68.87	82.05	14.66	75.59	87.05	18.15
20	72.27	68.89	82.13	14.89	75.64	86.98	18.08
Label Codec Interpol., sm Sec. 1.3							
Nearest-neighbors	67.12	62.54	78.25	12.66	71.70	84.68	18.96
Bicubic	71.55	67.95	80.64	15.30	75.16	87.95	18.60
Bilinear	71.79	68.40	81.42	15.23	75.18	86.89	18.69

3 Additional Qualitative Results

In this section of the supplemental material, we report additional qualitative results on label propagation.

In Fig. 3 we report qualitative results on our novel joint tracking and keypoint propagation method for real-world and synthetic pigeons (for quantitative results cf. Fig. 4 in our main paper). Please also check out our videos for further insights into this part. Videos can be found at https://urs-waldmann.github.io/improving-unsupervised-label-propagation/. In Fig. 4 we report qualitative results for joint tracking and keypoint propagation on JHMDB-test (for quantitative results cf. Tab. 3 in our main paper). In Fig. 5 we report qualitative results for O-VOS on DAVIS2017val (for quantitative results cf. Tab. 4).



Real-world sequence

Synthetic sequence

Fig. 3. *Qualitative results of our novel joint tracking and keypoint propagation method.* The first frame annotation is shown as the initial frame for each row with the corresponding ground-truth annotations. The following frames are sampled with even spacing from the remaining video. The second and fourth row show full frames of the real-world and synthetic sequences respectively. The first and third row show enlarged crops from these full frames to improve the visibility of the keypoints. The green box in the full frames indicates the cropping location. The red lines of the skeleton are the left side, blue lines the right side and magenta the connection to the beak.

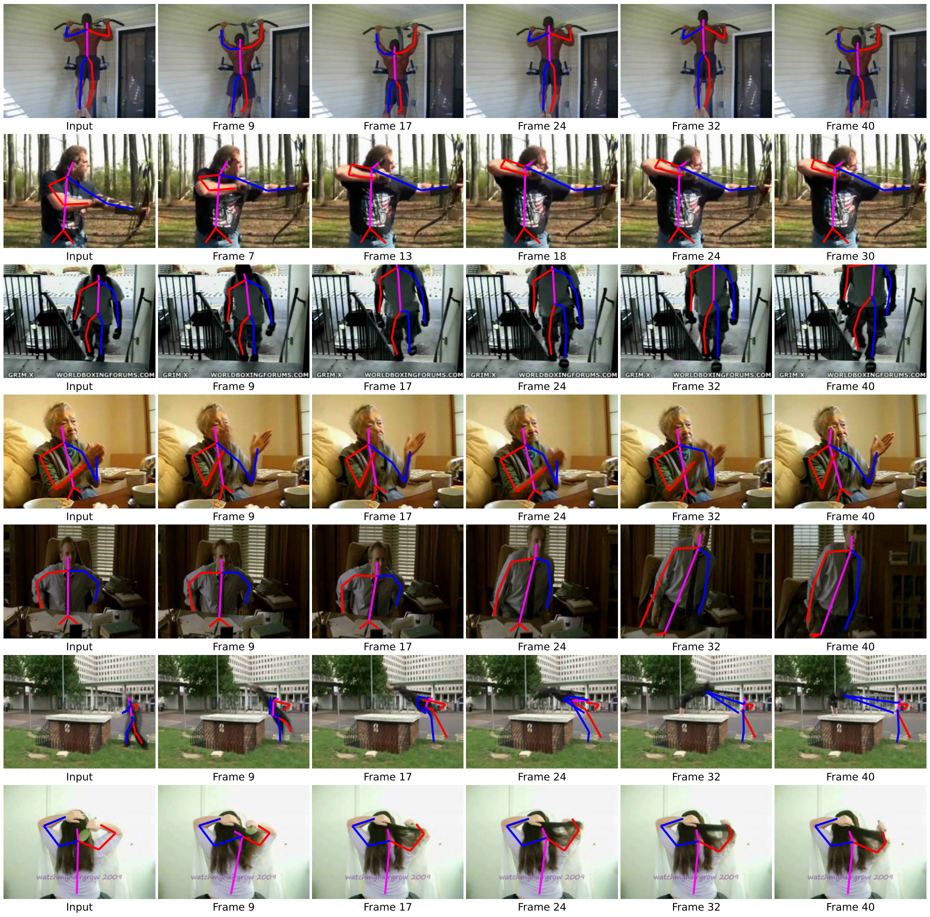


Fig. 4. *Qualitative results on JHMDB-test.* The index number and category of the shown videos are 35 (*pullup*), 186 (*shoot bow*), 199 (*climb stairs*), 205 (*clapping*), 47 (*stand*), 73 (*jump*) and 218 (*brush hair*), listed from top to bottom. The first four rows show videos where the pose tracking works as intended. The latter three rows are various error cases.

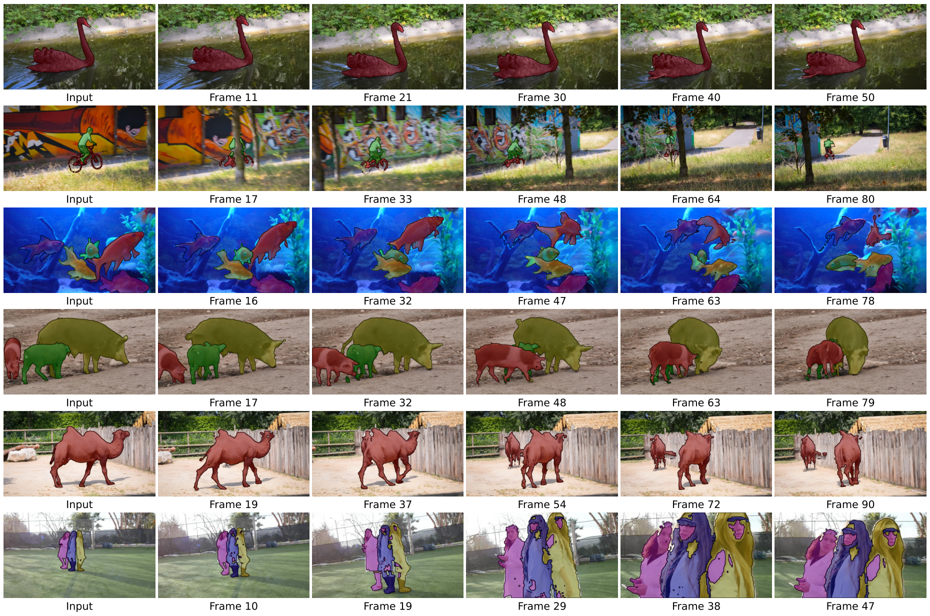


Fig. 5. *Qualitative results on DAVIS2017val with one-shot inference*. The shown sequences are *blackswan*, *bmx-trees*, *gold-fish*, *pigs*, *camel* and *lab-coat*, listed from top to bottom. The first frame shows the ground-truth mask used for initialization. The following frames were sampled over the entire video length maintaining even spacing in between.

4 Additional Quantitative Results

In his section of the supplemental material, we report more quantitative results on label propagation.

On the left side of Tab. 4 we report O-VOS results on DAVIS2017val (for qualitative results cf. Fig. 5). On the right side of Tab. 4 instead we show quantitative results on SegTrackV2 for O-VOS (quantitative results for Z-VOS are shown in Tab. 5 in our main paper).

Table 4. Multi-object O-VOS results on DAVIS2017val (left) and SegTrackV2 (right). On DAVIS2017 we outperform or match the performance of other self-supervised methods. It improves on DINO without using a significantly different amount of compute resources, simply by ensuring accurate label encoding and decoding. When we increase the number of context frames from 7 to 20 we can improve the results even further (cf. Tab. 3). On SegTrackV2 we can match the performance of popular supervised and semi-supervised methods that were state of the art only a few years prior.

Method	$\mathcal{J}\&\mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{J}_r$	$\mathcal{F}_m$	$\mathcal{F}_r$
MuG [15]	54.4	52.6	57.4	56.1	58.1
UVC [13]	58.2	56.8	65.7	59.5	65.1
UVC-track [13]	58.9	57.7	67.1	60.0	65.7
VINCE [3]	60.4	57.9	66.2	62.8	71.5
MAST [11]	65.5	63.3	73.2	67.6	77.7
STC [6]	67.6	64.8	76.1	70.2	82.1
STC-adapt [6]	68.3	65.5	78.6	71.0	82.9
DINO (ViT-B/8) [2]	71.4	67.9	80.7	74.8	**87.8**
Ours	71.8	68.4	81.4	75.2	86.9
Ours + context=20	**72.3**	**68.9**	**82.1**	**75.6**	87.0

Method	$\mathcal{J}_m$
BVS [16]	58.4
OSVOS [1]	65.4
MaskTrack [17]	70.3
RGMP [22]	71.1
MaskRNN [5]	72.1
LucidTracker [9]	77.6
Ours	78.0
Ours + context=20	**78.1**
STV† [20]	**78.1**

5 Details on the Synthetic Pigeon Dataset

The synthetic sequence is rendered with Blender and the Cycles engine. The 3D model is made of $40k$ polygons and textured using $4K$ images. Multiple actions like walking, eating, cleaning, and looking around are combined.

6 Rough Expected Inference Speed

Our inference speed on VOS has only negligible differences to DINO [2]. For keypoint propagation the inference speed of our method is similar to STC [6] because the same number of context frames is used and the remaining pipeline is very similar. The object tracking is beneficial to the inference speed because it allows to reduce the scale size. Tracking and scaling to 320 yields very similar result quality to keypoint propagation without tracking at 480 (cf. Tabs. 1 and 2) but is roughly three times as fast.

References

1. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR. pp. 221–230 (2017)
2. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
3. Gordon, D., Ehsani, K., Fox, D., Farhadi, A.: Watching the world go by: Representation learning from unlabeled videos. arXiv preprint arXiv:2003.07990 (2020)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
5. Hu, Y.T., Huang, J.B., Schwing, A.: Maskrnn: Instance level video object segmentation. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/6c9882bbac1c7093bd25041881277658-Paper.pdf
6. Jabri, A., Owens, A., Efros, A.: Space-time correspondence as a contrastive random walk. In: NeurIPS. pp. 19545–19560 (2020)
7. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: ICCV (2013)
8. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
9. Khoreva, A., Benenson, R., Ilg, E., Brox, T., Schiele, B.: Lucid data dreaming for object tracking. The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops (2017)
10. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: NeurIPS (2011)
11. Lai, Z., Lu, E., Xie, W.: Mast: A memory-augmented self-supervised tracker. In: CVPR (2020)
12. Lai, Z., Xie, W.: Self-supervised learning for video correspondence flow. In: BMVC (2019)
13. Li, X., Liu, S., De Mello, S., Wang, X., Kautz, J., Yang, M.H.: Joint-task self-supervised learning for temporal correspondence. In: NeurIPS (2019)
14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755 (2014)
15. Lu, X., Wang, W., Shen, J., Tai, Y.W., Crandall, D.J., Hoi, S.C.H.: Learning video object segmentation from unlabeled videos. In: CVPR (2020)
16. Märki, N., Perazzi, F., Wang, O., Sorkine-Hornung, A.: Bilateral space video segmentation. In: CVPR. pp. 743–751 (2016)
17. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: CVPR. pp. 2663–2672 (2017)
18. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: ImageNet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
19. Wang, N., Zhou, W., Song, Y., Ma, C., Liu, W., Li, H.: Unsupervised deep representation learning for real-time tracking. IJCV **129**, 400–418 (2021)
20. Wang, W., Song, H., Zhao, S., Shen, J., Zhao, S., Hoi, S.C.H., Ling, H.: Learning unsupervised video object segmentation through visual attention. In: CVPR. pp. 3064–3074 (2019)

21. Wang, X., Jabri, A., Efros, A.A.: Learning correspondence from the cycle-consistency of time. In: CVPR (2019)
22. Wug Oh, S., Lee, J.Y., Sunkavalli, K., Joo Kim, S.: Fast video object segmentation by reference-guided mask propagation. In: CVPR. pp. 7376–7385 (2018)