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

Supplemental Material

Liulei Li[1,4]*, Wenguan Wang[1]†, Tianfei Zhou[2], Jianwu Li[3], Yi Yang[1]

[1] ReLER, CCAI, Zhejiang University [2] ETH Zurich [3] Beijing Institute of Technology [4] Baidu VIS

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

In this document, we first provide the pseudo code of our approach in §A. Then, additional analyses with respect to pseudo mask updating and recurrent refinement are presented in §B and §C, respectively. Last, §E offers additional qualitative segmentation results.

A. Pseudo Code

The inference mode of our method is summarized in Alg. S1. Note that the recurrent refinement procedure is included.

B. Analysis of Pseudo Mask Updating

During training, our method conducts online space-time clustering to progressively refine pseudo masks with gradually improved visual representations. Fig. S1-S4 provide qualitative analysis of this strategy on YouTube-VOS [1] `train`. Here, 'Initial' corresponds to the pseudo masks created right after correspondence learning, while 'Final' refers to the masks that are obtained after nine online updates (once per 10 epochs from epoch 300 to 400). The first row shows the clustering results and the second row gives the pseudo masks derived from the clustering results. We can see that 1) our correspondence learning can indeed provide meaningful features for reliable clustering, leading to satisfactory initial pseudo labels, and 2) the pseudo masks are continuously improved with online updating, *e.g.*, background are suppressed and foreground are progressively highlighted and more spatiotemporally consistent.

C. Analysis of Recurrent Refinement

In Fig. S5, we further analyze visual effects of recurrent refinement over three representative sequences on DAVIS$_{17}$ `val`. For *Round 0*, we directly leverage $\boldsymbol{V}_q$ (Eq. 7) for mask decoding. For *Round 1*, the segmentation results (*i.e.*, $\hat{Y}_q$) are produced following Eq. 9. For *Round 2*, we replace $\overline{Y}_q$ in Eq. 9 by $\hat{Y}_q$ (in *Round 1*) and subsequently conduct mask decoding to yield refined masks. It can be observed that the segmentation quality is progressively improved with iterative refinement, consistent with the results in Table 5e.

Algorithm S1 Pseudo-code for the inference mode of our approach in a PyTorch-like style

```
# I_q: query frame
# I_r: reference frames
# Y_r: reference masks of I_r
# R: number of round for recurrent refinement
# N: number of reference frames

def visual_encoder(I):
   res4, res3, res2 = BACKBONE(I)
   key = MLP(res4)
   key = normalize(key)
   return key, res4, res3, res2

def mask_encoder(I, Y):
   res4, _, _ = BACKBONE([I, Y])
   value = MLP(res4)
   return value

def inference(I_q, I_r, Y_r, R=2):
   # NHW x D'
   V_r = mask_encoder(I_r, Y_r)
   # NHW x D
   K_r, _, _, _ = visual_encoder(I_r)
   # HW x D
   K_q, res4, res3, res2 = visual_encoder(I_q)

   #===== compute the affinity (Eq.6) ======#
   # NHW x HW
   A = mm(K_r, K_q.transpose())
   A = softmax(A)

   #=== assemble support features (Eq.7) ===#
   # HW x D'
   V_q = mm(A.transpose(), V_r)

   #==== compute the coarse mask (Eq.8) ====#
   # HW x 1
   Y_q = mm(A.transpose(), Y_r)

   #======== recurrent refinement ==========#
   for _ in range(R):
   #===== predict segmentation (Eq.9) ======#
     V_q_overline = mask_encoder(I_q, Y_q)
     V_q_new = cat([V_q, V_q_overline], dim=0)
     Y_q = DECODER(V_q_new, res3, res2)

   return Y_q
```

`mm`: matrix multiplication; `normalize`: ℓ_2 normalization;
`cat`: concatenation; `softmax`: row-wise softmax.

*Work done during an internship at Baidu VIS.

†Corresponding author: *Wenguan Wang*.

Method	TimeCycle [2]	CRW [4]	CLTC [8]	VFS [5]	LIIR [6]	Ours
mIoU	28.9	38.6	37.8	39.9	41.2	**42.9**

Table S1. Quantitative results on VIP [7] `test`.

Method	Training time (Min/Epoch)	$\mathcal{J}\&\mathcal{F}_m\uparrow$
MAST	26.8	65.5
MAST + $\mathcal{L}_{\text{Seg}}$	28.7	69.0
CRW	223.8	67.6
CRW + $\mathcal{L}_{\text{Seg}}$	239.9	71.8
$\mathcal{L}_{\text{Corr}}$ + (ours)	5.1	68.8
$\mathcal{L}_{\text{Corr}}$ + $\mathcal{L}_{\text{Seg}}$	5.5	74.5

Table S2. Analysis of training speed on DAVIS$_{17}$ [9] `val`.

D. Additional Application Task

We additionally test our model on the task of body part propagation. Following [2–6], we conduct experiment on VIP [7] benchmark dataset. It can be seen in Table S1 that our method achieves the best performance.

E. Additional Qualitative Results

We provide more comparison results on DAVIS$_{17}$ [9] `val` in Fig. S6-S7 and YouTube-VOS [1] `val` in Fig. S8-S9, respectively. We can find that our approach suffers less from error accumulation over time, and yields consistently better results against other competitors.

F. Training Time

The comparisons of training time are summarized in Table S2. All experiments are conducted on one Tesla A100 GPU with `ResNet-18` backbone. $\mathcal{L}_{\text{Seg}}$ is involved in optimization after 300 training epochs. It can be seen that our method brings only slight training speed delay (around 8%), while offering remarkable performance improvement.

G. Limitation Discussion

Currently we directly leverage the k-means algorithm to cluster pixels. The k-means clustering, though simple, is less efficient compared with some more advanced ones, such as [10, 11] which consider clustering from the perspective of optimal transport. We leave this as a part of our future work.

References

[1] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 1, 2, 3, 4, 5, 6, 7, 10, 11

[2] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 2

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

[4] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 2, 8, 9, 10, 11

[5] Jiarui Xu and Xiaolong Wang. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. In *ICCV*, 2021. 2

[6] Liulei Li, Tianfei Zhou, Wenguan Wang, Lu Yang, Jianwu Li, and Yi Yang. Locality-aware inter-and intra-video reconstruction for self-supervised correspondence learning. In *CVPR*, 2022. 2, 8, 9, 10, 11

[7] Qixian Zhou, Xiaodan Liang, Ke Gong, and Liang Lin. Adaptive temporal encoding network for video instance-level human parsing. In *ACM MM*, 2018. 2

[8] Sangryul Jeon, Dongbo Min, Seungryong Kim, and Kwanghoon Sohn. Mining better samples for contrastive learning of temporal correspondence. In *CVPR*, 2021. 2

[9] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 2, 7, 8, 9

[10] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 2

[11] Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. In *ICLR*, 2018. 2

[12] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *CVPR*, 2020. 8, 9, 10, 11

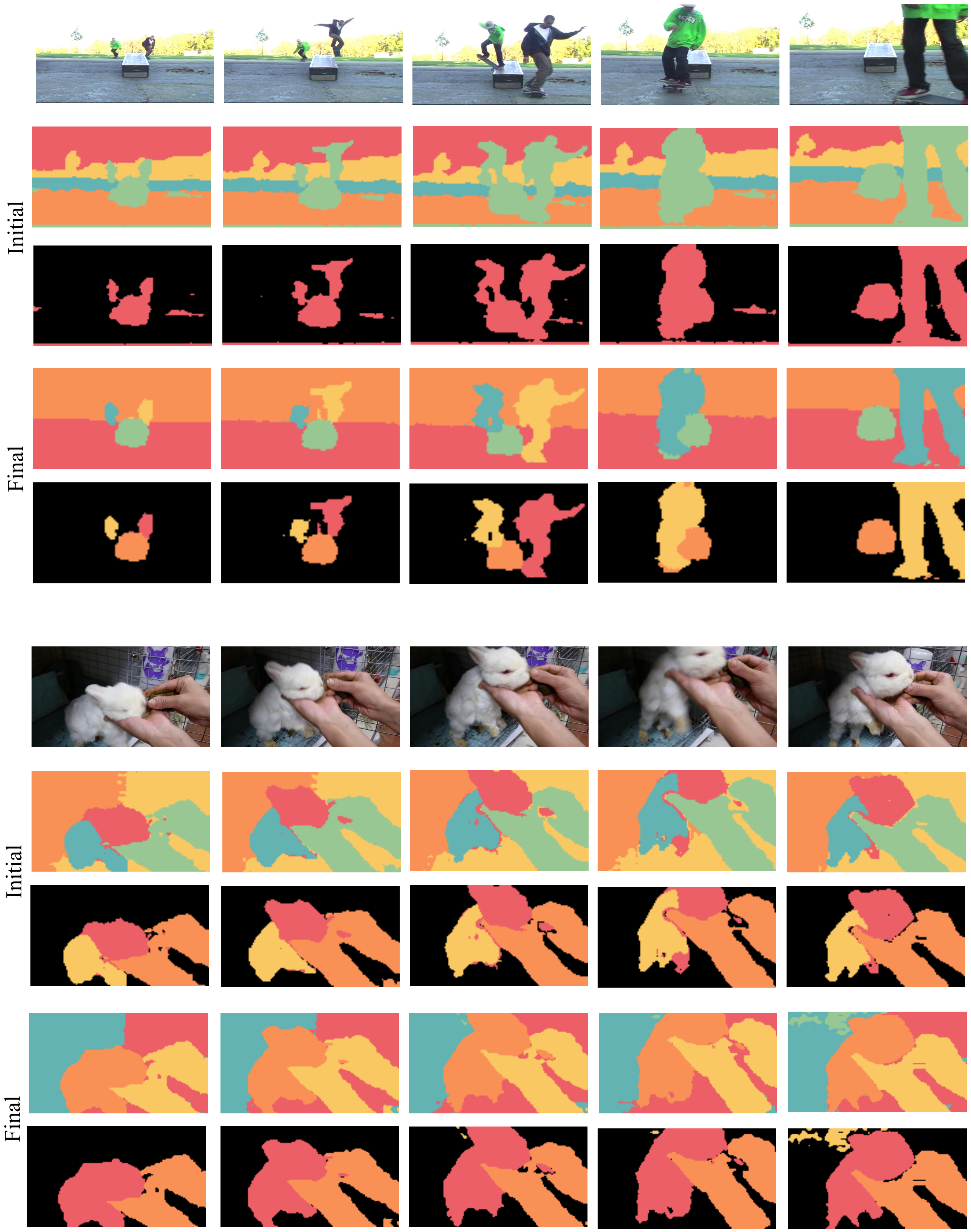

Figure S1. **Qualitative analysis of pseudo mask generation and update** on YouTube-VOS [1] `train`. 'Initial': masks created right after correspondence learning; 'Final': masks obtained after nine online updates (once per 10 epochs from epoch 300 to 400). The first row shows the clustering results and the second row gives the pseudo masks derived from the clustering results.

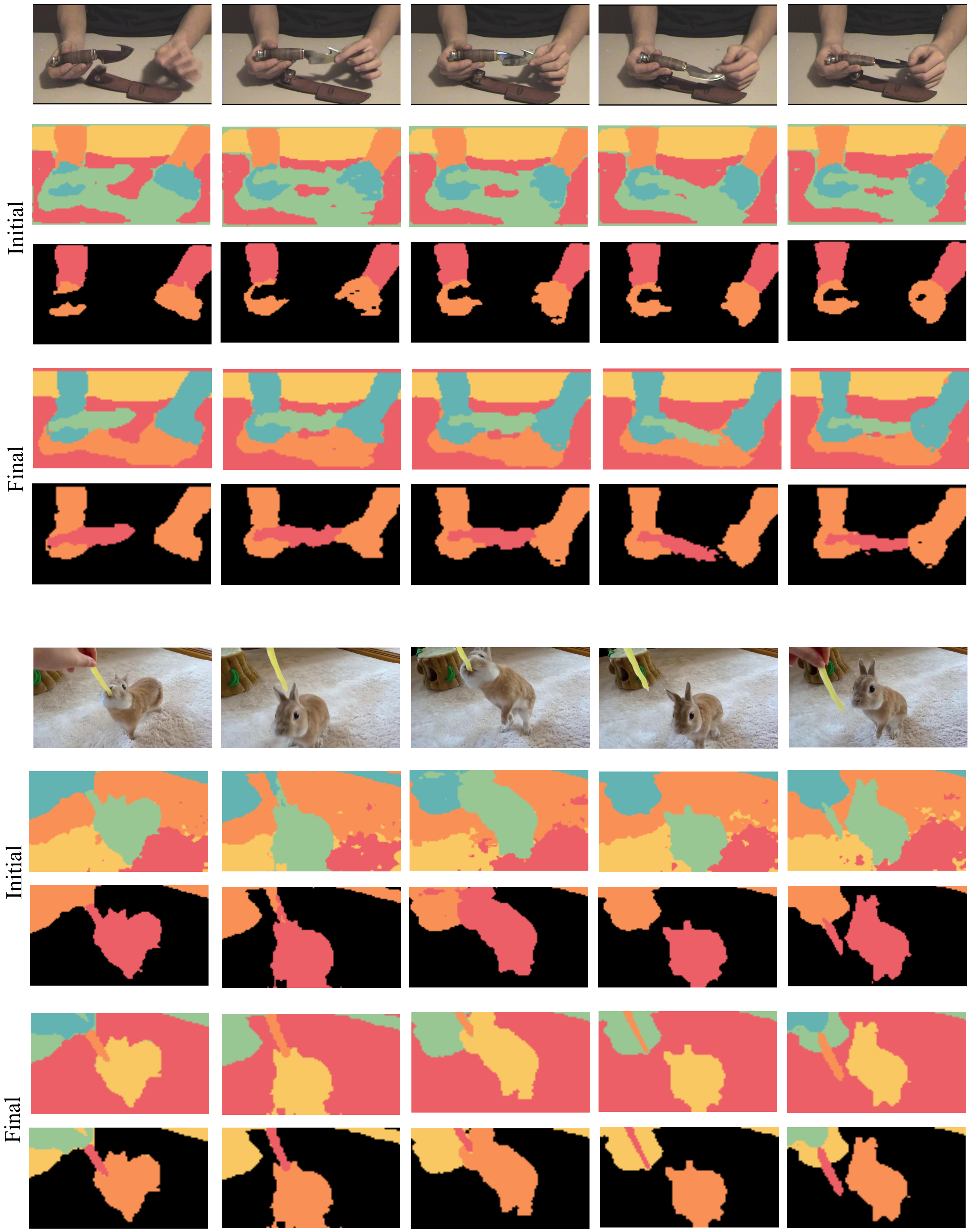

Figure S2. **Qualitative analysis of pseudo mask generation and update** on YouTube-VOS [1] `train`. 'Initial': masks created right after correspondence learning; 'Final': masks obtained after nine online updates (once per 10 epochs from epoch 300 to 400). The first row shows the clustering results and the second row gives the pseudo masks derived from the clustering results.

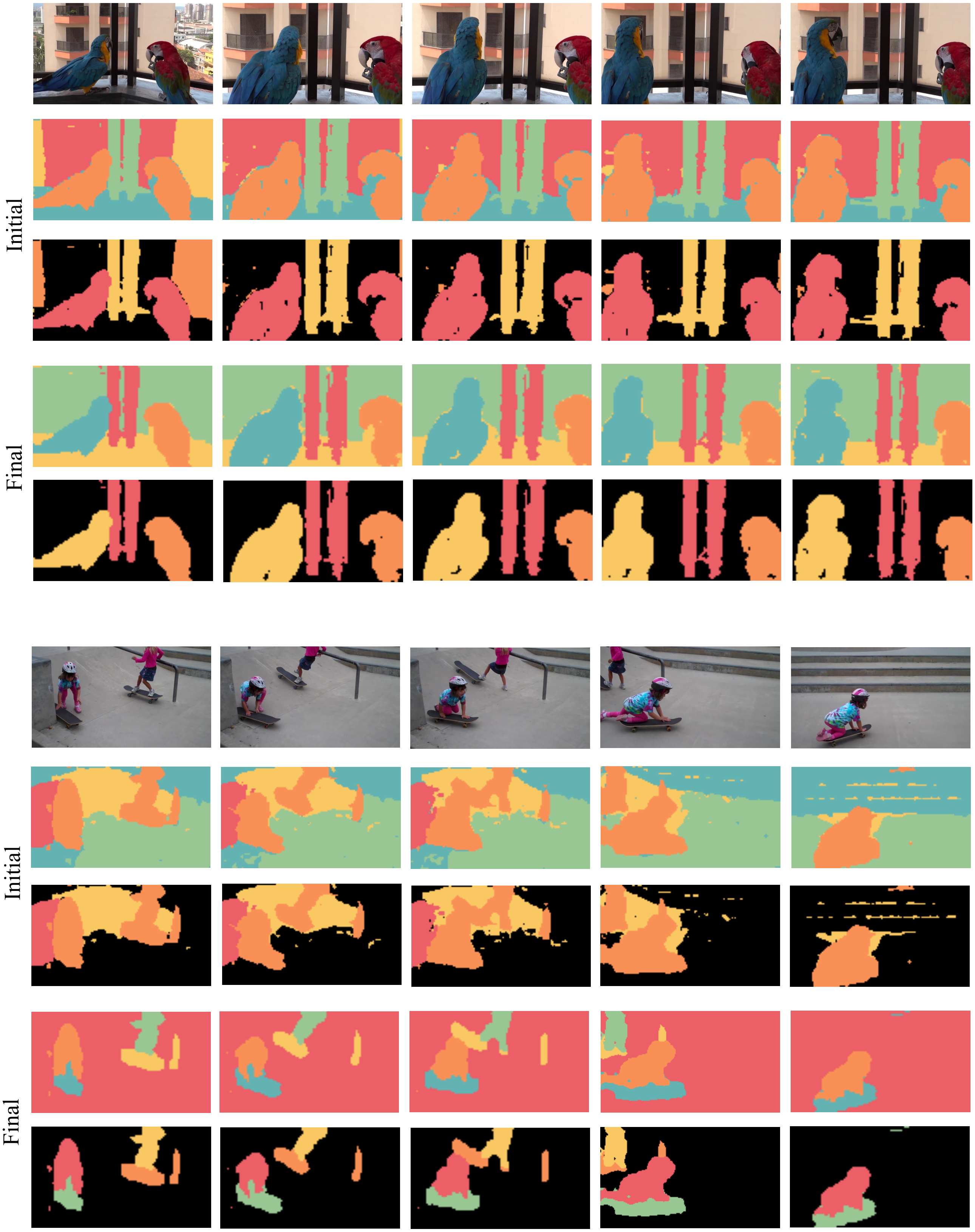

Figure S3. **Qualitative analysis of pseudo mask generation and update** on YouTube-VOS [1] `train`. ‘Initial’: masks created right after correspondence learning; ‘Final’: masks obtained after nine online updates (once per 10 epochs from epoch 300 to 400). The first row shows the clustering results and the second row gives the pseudo masks derived from the clustering results.

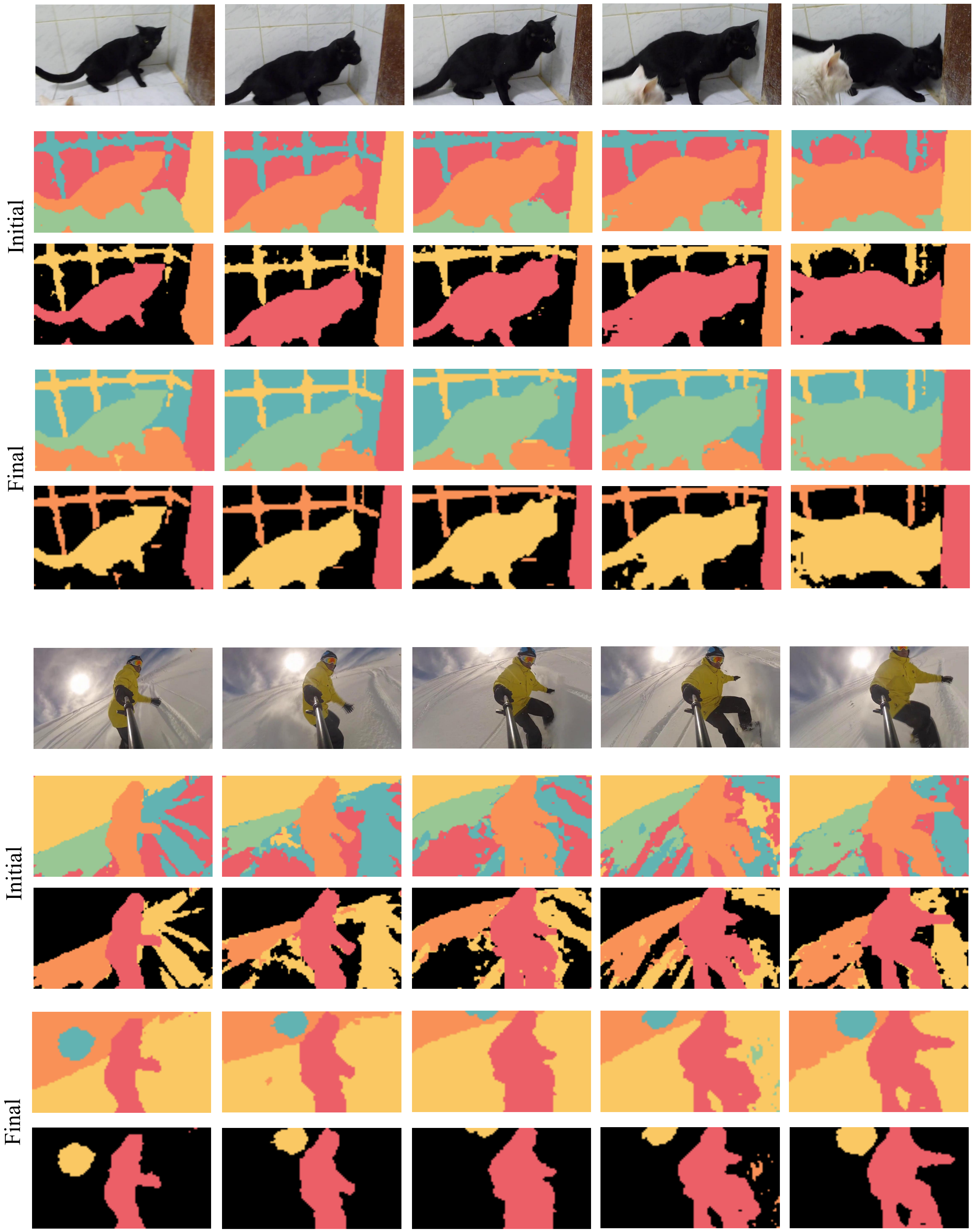

Figure S4. **Qualitative analysis of pseudo mask generation and update** on YouTube-VOS [1] `train`. ‘Initial’: masks created right after correspondence learning; ‘Final’: masks obtained after nine online updates (once per 10 epochs from epoch 300 to 400). The first row shows the clustering results and the second row gives the pseudo masks derived from the clustering results.

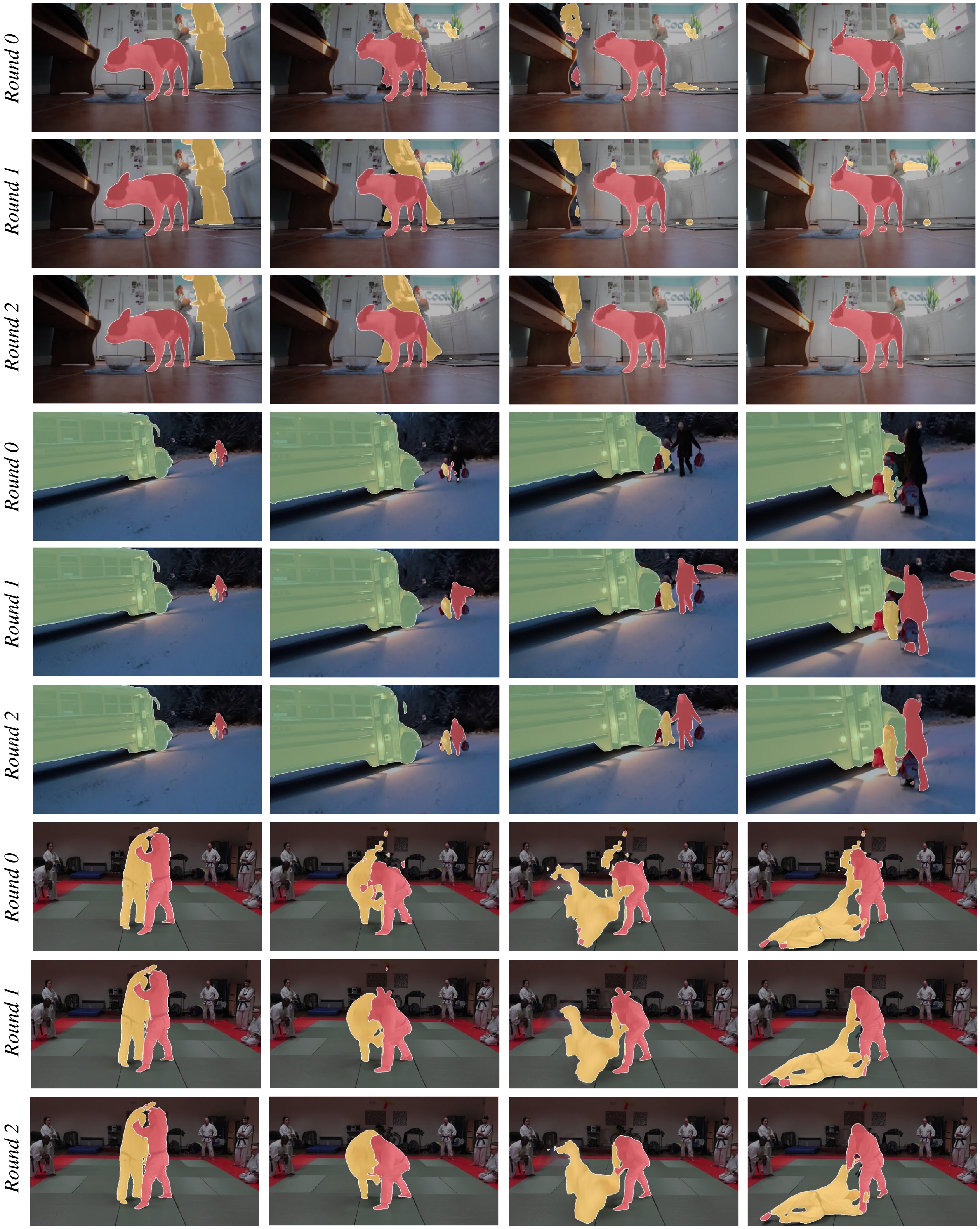

Figure S5. **Qualitative analysis of recurrent refinement** on DAVIS$_{17}$ [9] `val` and YouTube-VOS [1] `val`. For *Round 0*, we directly leverage $\boldsymbol{V}_q$ (Eq. 7) for mask decoding. For *Round 1*, the segmentation results (*i.e.*, $\hat{Y}_q$) are produced following Eq. 9. For *Round 2*, we replace $\overline{Y}_q$ in Eq. 9 by $\hat{Y}_q$ (in *Round 1*) and subsequently conduct mask decoding to yield refined masks.

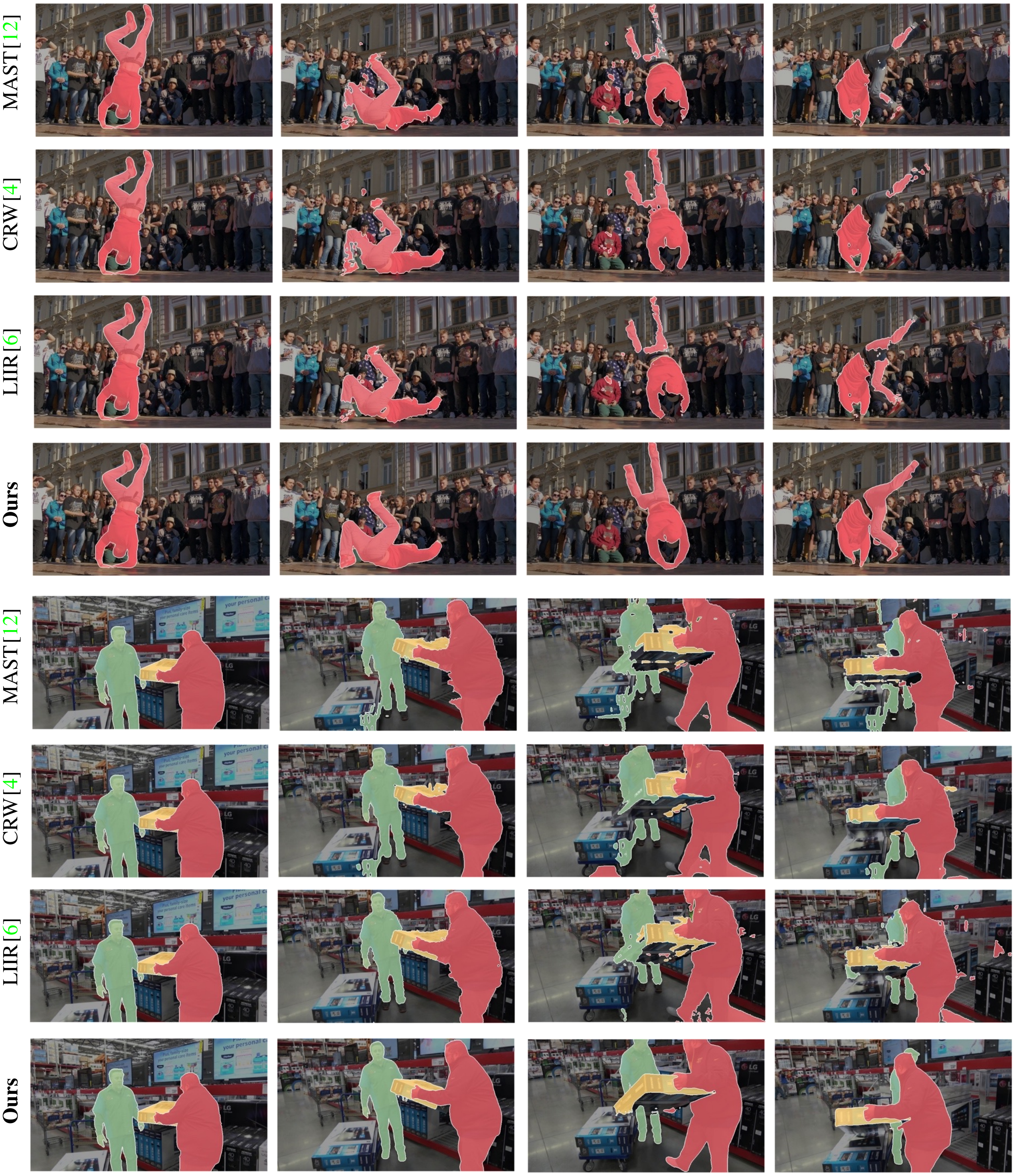

Figure S6. **Visual comparison results** on DAVIS$_{17}$ [9] `val`.

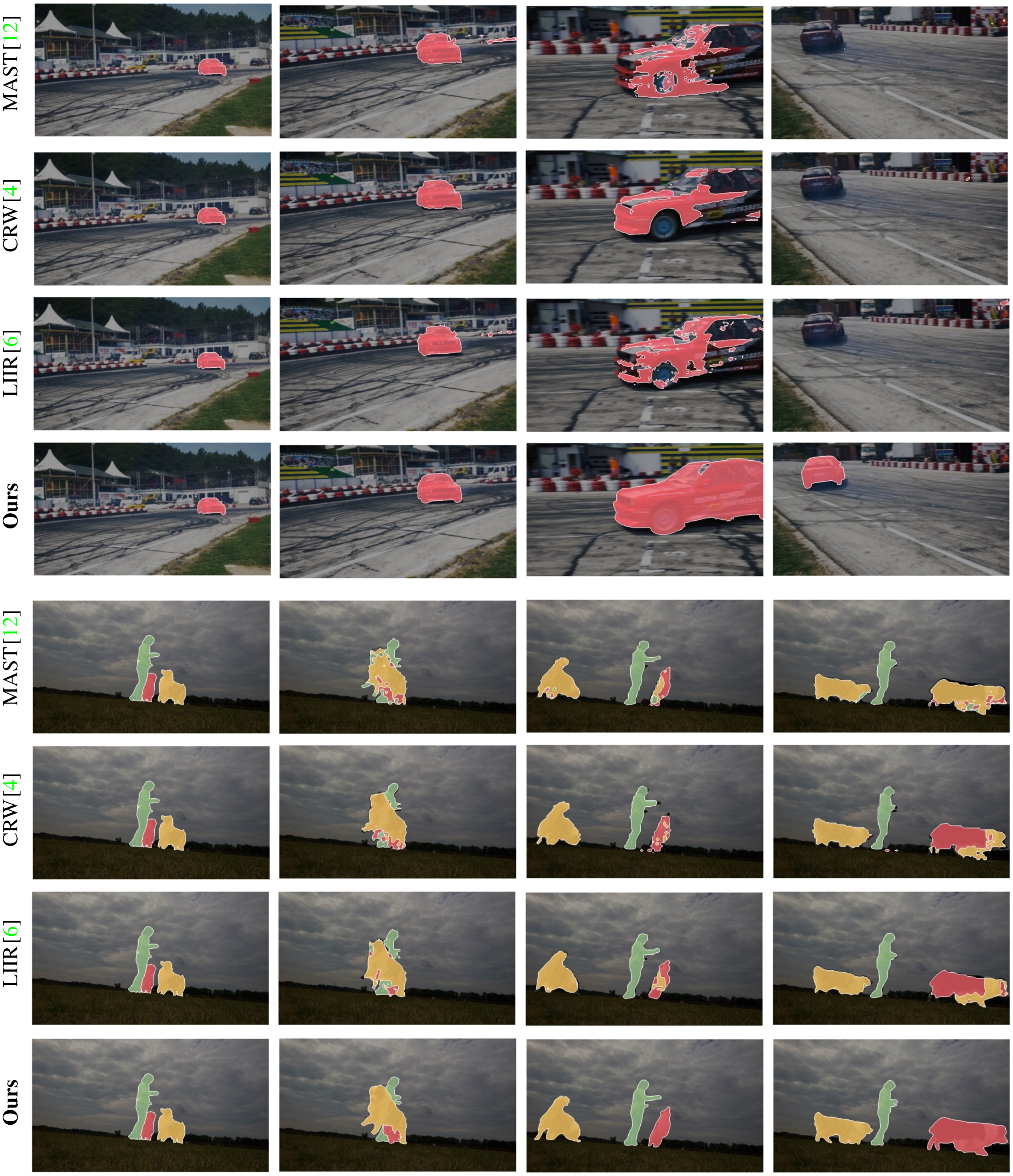

Figure S7. **Visual comparison results** on DAVIS$_{17}$ [9] `val`.

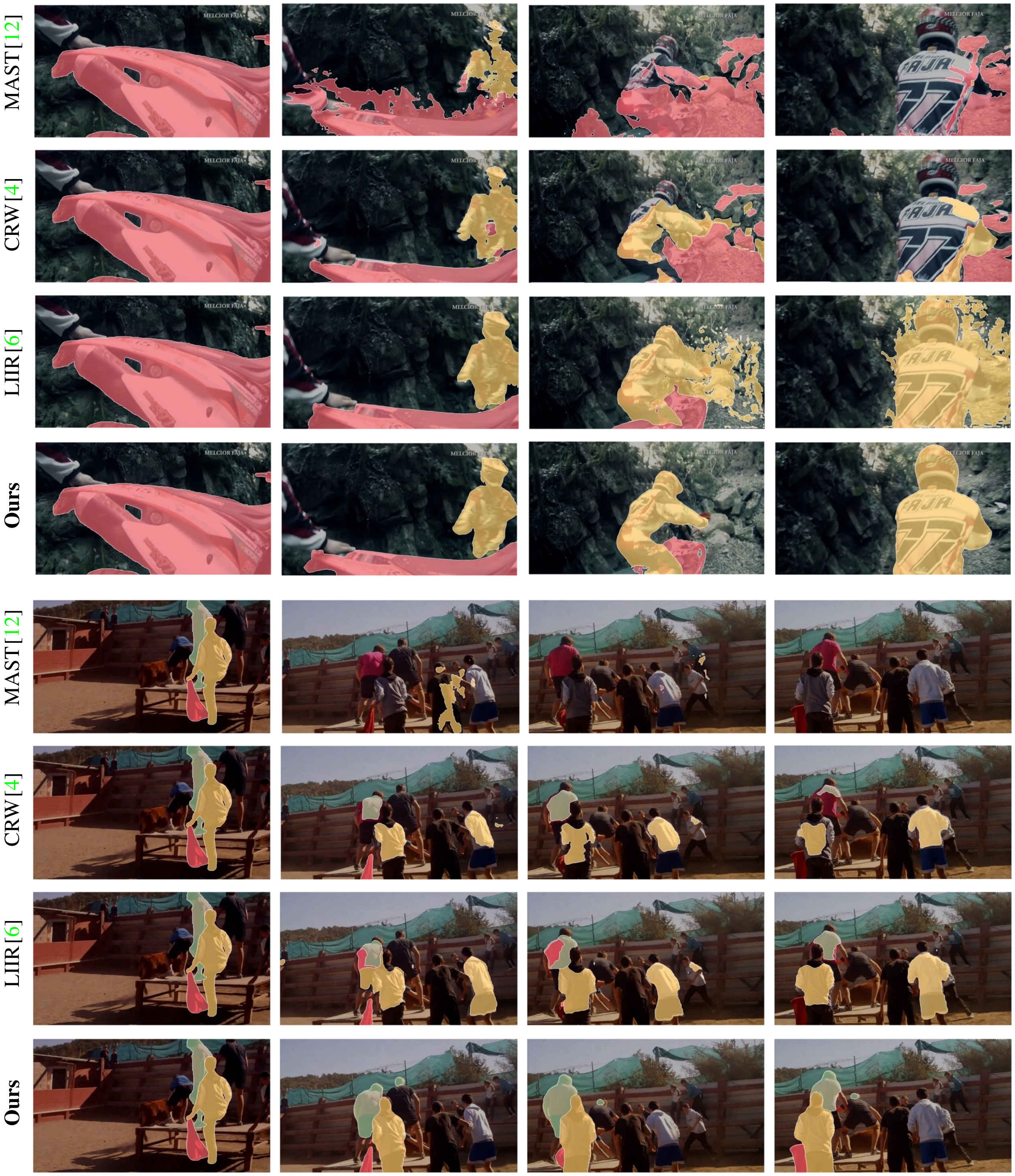

Figure S8. **Visual comparison results** on YouTube-VOS [1] `val`.

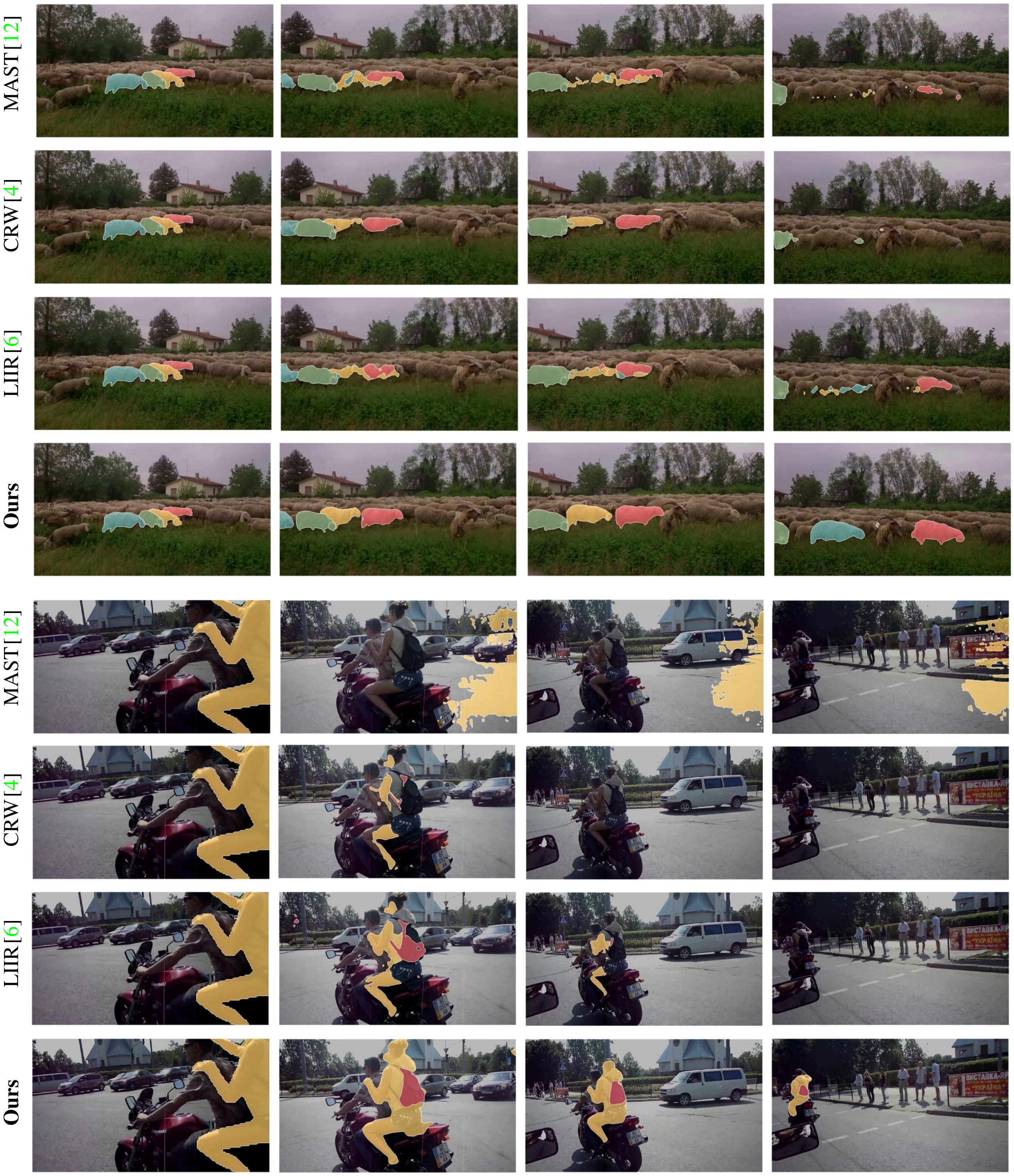

Figure S9. **Visual comparison results** on YouTube-VOS [1] `val`.