UniVS: Unified and Universal Video Segmentation with Prompts as Queries

Minghan Li[1,2*], Shuai Li[1,2*], Xindong Zhang[2] and Lei Zhang[1,2†]
[1]The Hong Kong Polytechnic University [2]OPPO Research Institute
liminghan0330@gmail.com, xindongzhang@foxmail.com, {csshuaili, cslzhang}@comp.polyu.edu.hk

In this supplementary file, we provide the following materials:

A. Datasets and Evaluation Metrics

Video segmentation (VS) tasks can be divided into two groups: category-specified and prompt-specified VS tasks. Table 1 summarizes the statistics of different VS datasets.

A.1. Category-specified VS Datasets

Category-specified VS tasks include video instance segmentation (VIS) [17, 23], video semantic segmentation (VSS) [13] and video panoptic segmentation (VPS) [7, 12], where the object categories need to be specified.

Video Instance Segmentation (VIS) involves identifying and segmenting individual objects within each frame of a video while maintaining temporal consistency across frames. There are two large-scale VIS datasets: YouTube-VIS [23] series and OVIS [17]. **YouTube-VIS** [23] has three versions: YT19/21/22. The commonly used version is YT21, which contains 2,985 training, 421 validation, and 453 test videos over 40 'thing' categories. The number of frames per video is between 19 and 36. **OVIS** [17] targets at distinguishing occluded objects in long-time videos (up to 292 frames), which includes 607 training, 140 validation, and 154 test videos, scoping 25 'thing' categories. VIS task adopts average precision (AP_*), average recall (AR_*) and the mean value of AP (mAP) as metrics for evaluation.

Video Semantic Segmentation (VSS) needs to perform pixel-level labeling of semantic categories in each frame of a video. **VSPW** [13] is the first large-scale video scene parsing dataset, containing 3,536 annotated videos and 124 semantic thing/stuff classes. VSS uses mIoU, mVC_8 and mVC_{16} as metrics for evaluation, where mean video consistency (mVC_*) evaluates the category consistency among long-range adjacent frames ('*' indicates the number of frames in a video clip).

Video Panoptic Segmentation (VPS) combines VIS and VSS tasks by simultaneously identifying and tracking individual object instances while assigning semantic labels to each pixel. The goal is to achieve a comprehensive understanding of both instance-level and semantic-level information across the video sequence. **VIPSeg** [12] is the first large-scale VPS dataset in the wild, which shares the original videos from the VSPW dataset. VIPSeg has pixel-level panoptic annotations, covering a wide range of real-world scenarios and categories. There are two commonly used evaluation metrics for the VPS task: VPQ [7] and STQ [19]. Video Panoptic Quality (VPQ) computes the average mask quality by performing tube IoU matching across a small span of frames. Segmentation and Tracking Quality (STQ) is proposed to measure the segmentation quality and long term tracking quality simultaneously.

A.2. Prompt-specified VS Datasets

Prompt-specified VS focuses on identifying and segmenting specific targets throughout the video, where visual prompts or textual descriptions of the targets need to be provided. It includes video object segmentation (VOS) [15], panoptic VOS (PVOS) [21] and referring VOS (RefVOS) [18].

Video Object Segmentation (VOS) segments a particular object throughout the entire video given only the object mask at the first frame, which can be viewed as the extension of interactive segmentation from spatial to temporal dimension. **DAVIS** [16], an early proposed VOS dataset, contains a total of 90 videos. **YouTube-VOS** (YT18) [20] consists of 4,453 short video clips with 94 different object categories. **MOSE** [4] targets at complex video object segmentation, whose videos partially inherit from OVIS [17]. MOSE contains 2,149 video clips and 36 object categories. To evaluate the performance, region jaccard J and countour accuracy F are computed for 'seen' and 'unseen' classes separately, denoted by subscripts s and u. G^{th} is the average ($J\&F$) over both seen and unseen classes.

Panoptic VOS (PVOS) extends the above VOS task by taking stuff classes into account. Based on the VIPSeg dataset, **VIPOSeg** [21] is developed for PVOS. It contains exhaustive object annotations and covers various real-world

[*] Equal contribution, † Corresponding author.

Tasks	VIS			VSS	VPS	PVOS	VOS				RefVOS	
Datsets	YT19	YT21	OVIS	VSPW	VIPSeg	VIPOSeg	DAVIS	YT18	MOSE	BURST	RefDAVIS	RefYT
Videos	2.8k	3.8k	1.0k	3.5k	3.5k	3.5k	0.09k	4.4k	2.1k	1.9k	0.09k	4.0k
Images	97k	92k	51k	252k	85k	85k	6k	97k	96k	196k	6k	97k
Masks	131k	232k	296k	-	926k	926k	13k	197k	431k	600k	13k	197k
Classes	40	40	25	124	124	124	-	94	36	482	-	-
Expressions	-	-	-	-	-	-	-	-	-	-	1.5k	28k
Thing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Stuff	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗
Exhaustive	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗

Table 1. Statistics of different video segmentation datasets. The datasets labeled with the same color share the source video data but have different annotation formats, such as VSPW, VIPSeg and VIPOSeg.

Datasets		Images				Videos							Settings			
		IS	PS	IS	Ref	VIS	VIS	VPS	VOS	VOS	VOS	Ref				
Training	Frames	SA1B	COCO	LVIS	RefCOCO	YT21	OVIS	VIPSeg	YT18	MOSE	BURST	RefYT	GPUs	Lr	Max Iter	Step
Stage 1	1	1.0	1.0	0.5	1.0	-	-	-	-	-	-	-	16	1e-4	354k	342k
Stage 2	3	0.25	0.5	0.25	0.35	0.25	0.35	0.5	0.25	0.25	0.25	0.35	8	5e-5	708k	684k
Stage 3	5-7	0.25	0.5	0.25	0.35	0.25	0.35	0.5	0.25	0.25	0.25	0.35	8	5e-5	177k	162k

Table 2. Implementation details in training. The sampling weights of each dataset during different training stages are given. '-' means that the dataset is not used. 'Step' means the iterations when the learning rate is reduced.

object categories, which are carefully divided into subsets of thing/stuff and seen/unseen classes for comprehensive evaluation. This newly proposed benchmark uses eight separate metrics, including four mask IoUs for seen/unseen thing/stuff and four boundary IoUs [1] for seen/unseen thing/stuff, respectively. The overall performance $G^{th\&sf}$ is the average of these eight metrics.

Referring VOS (RefVOS) aims to segment the target object in a video based on the natural language description, which is a challenging multi-modal segmentation task. **RefDAVIS** and **RefYT** [18] are two RefVOS datasets based on DAVIS and YouTube-VOS [20], respectively. RefYT is a large-scale benchmark covering 3,978 videos with around 28K language descriptions. The evaluation metrics include region similarity (J), contour accuracy (F) and their average value ($J\&F$).

B. Training and Inference Details

B.1. Training Losses

There are three terms in the training loss:

$$L = \lambda_{\text{mask}} L_{\text{mask}} + \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{reid}} L_{\text{reid}}, \tag{1}$$

where $\lambda_{\text{mask}}, \lambda_{\text{cls}}$ and λ_{reid} are the hyper-parameters to balance the multiple loss terms. Their default values are set to 5, 3, 0.5, respectively. During training, mask annotations of all VS tasks are fully utilized to train the learnable and prompt queries.

Mask Loss contains two common functions: Dice loss [3] and Binary Cross-Entropy (BCE) loss. It can be formulated as follows:

$$L_{\text{mask}} = \sum\nolimits_{t=1}^{T} L_{\text{mask}}(M^t, \bar{M}^t) + L_{\text{mask}}(M^{*t}, \bar{M}^t),$$

where M^t, M^{*t} are the matched masks for learnable queries and prompt queries, respectively, and $\bar{M}^t$ denotes the ground-truth mask. t and T are the frame index and the number frames of the input video clip.

Classification Loss only applies to category-specified VS tasks. We leverage the similarity between query embeddings and CLIP embeddings of category names for recognition. The classifier S can be obtained by:

$$S = 1/T \ \times \ \text{Cosine}(f_{\text{cls}}([\mathbf{q}, \mathbf{q}^*]),\ P_{cate}),$$

where P_{cate} is the text embedding of category names produced by CLIP text encoder, f_{cls} converts query embeddings from the visual space to the language space using an MLP layer. T is a temperature to amplify the logit. We employ focal loss [10] to supervise the classifier.

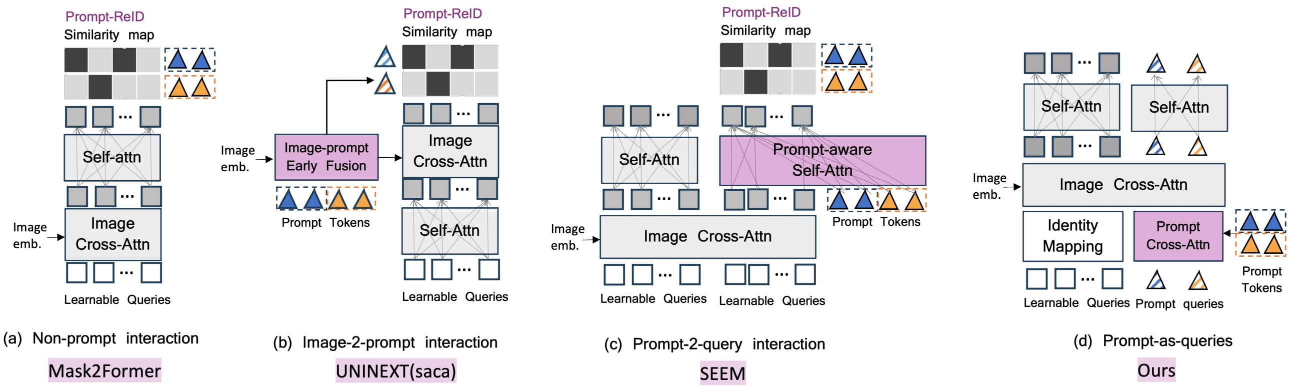
ReID Loss aims to maintain the temporal consistency in VS tasks, which can be formulated as:

$$L_{\text{ReID}} = L_{\text{ReID}}(\mathbf{q}, \mathbf{q}) + L_{\text{ReID}}(\mathbf{q}, \mathbf{q}^*) + L_{\text{ReID}}(\mathbf{q}^*, \mathbf{q}^*),$$

where the second term aims to align prompt queries and learnable queries within the same feature space. We utilize the contrastive loss and the auxiliary loss proposed in [14] for the ReID loss.

B.2. Training Stages

The whole training process consists of three consecutive stages: image-level joint training, video-level joint training and long video finetuning. In the first stage, we jointly pretrain UniVS on multiple image datasets, including SA1B [8], COCO[11], LVIS [5], and the mixed dataset of RefCOCO[6], RefCOCO+[6], RefCOCOg. Due to limited computational resources, we randomly select 250k images from the 1M images (2.5%) in the original SA1B



(a) Non-prompt interaction
Mask2Former

(b) Image-2-prompt interaction
UNINEXT(saca)

(c) Prompt-2-query interaction
SEEM

(d) Prompt-as-queries
Ours

Figure 1. Architecture comparison of mask decoder layers in unified segmentation models, including Mask2Former [2], UNINEXT[22], SEEM[25] and our UniVS. These methods differ in image and prompt interaction. Note that feed-forward network (FFN) is omitted here.

dataset for training. It has been experimentally shown in [8, 9] that the performance of the model trained on 3% SA1B images is slightly lower than the model trained on the entire images. In the last two stages, UniVS is trained on image and video datasets, including YT21, OVIS, VIPSeg, YT18, MOSE, Burst and RefYT. Similar to UNINEXT [22], to avoid the model forgetting previously learned knowledge on image-level datasets, we generate pseudo video clips from image datasets and merge them into jointly training on video datasets.

In Table 2, we show the sampling weights of each dataset in each training stage, as well as the number of GPU (GPUs), learning rate (Lr), the maximum iterations (Max Iter) and the time to reduce the learning rate (Step). For UniVS with R50 backbone, the training time for stage 1/2/3 on 16/8/8 V100 GPUs is 9.7/7.5/3.6 days, respectively. And UniVS with Swin-T/B/L backbones need similar training times for stage 1/2/3 on 16/8/8 A100 GPUs.

C. More Ablation Studies

Except specifically stated, experimental results in this section are evaluated using the ResNet50 backbone.

C.1. Comparison of Unified Architectures

To show the superiority of our proposed unified video segmentation architecture, we compare UniVS with popular unified segmentation frameworks, including Mask2Former [2], UNINEXT [22], and SEEM [25]. The architecture comparison is illustrated in Fig. 1.

The original Mask2Former [2] can process multiple category-specified VS tasks, such as VIS/VSS/VPS, but cannot handle prompt-specified segmentation tasks, such as VOS/PVOS/RefVOS. UNINEXT [22] is an object-centric segmentation model, which aligns text prompts with image embeddings by introducing a vision-language early fusion module in the pixel decoder (see Fig. 1b). UNINEXT is built upon the DeformabelDETR [24] framework, which is more suitable for instance-level detection and segmentation, but exhibits relatively weaker performance in detecting and segmenting stuff entities. SEEM [25] is designed for image segmentation. It introduces an extra group of learnable queries and extends the keys and values of self-attention layers to integrate prompt information, as shown in Fig 1c. However, when multiple prompt entities are presented, SEEM needs to utilize a post-processing matching stage to locate the targets from all predicted masks.

It can be observed that previous unified architectures require back-end matching between prompt tokens and learnable queries to identify the targets, which is detrimental to maintain entity consistency across frames. In contrast, our UniVS transfers all VS tasks to the prompt-guided target segmentation to explicitly decode masks, and thus the matching strategy is only used when detecting newly appeared entities from learnable queries, as shown in Fig 1d.

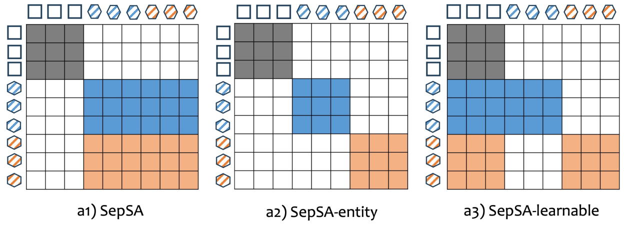
For quantitative performance comparison, we exclude UNINEXT[22] here, because it uses DefDETR [24] architecture instead of Mask2Former architecture, making it hard to be compared directly. We train the Mask2Former, SEEM and UniVS models using the same training settings and datasets (the first two stages in Sec B.2.). The results are shown in Table 3. While Mask2Former and SEEM may perform well on some of the VS tasks, UnivS performs the best on almost all VS tasks, demonstrating the superiority of our proposed architecture.

C.2. Inference Process

Separated Self-attention Types. As shown in Table 4(a), in the separated self-attention layer, we test three ways of interaction between learnable and prompt queries. Specifically, 'SepSA' refers to separate self-attention calculations for learnable and prompt queries respectively. 'SepSA-entity' involves interaction among prompt queries belonging to the same entity, with no visibility across different

Video Tasks	VIS		VPS		VOS	RefVOS
Method	YT21	OVIS	VIPSeg		YT18	RefYT
	mAP	mAP	VPQ	STQ	G^{th}	J&F
Mask2Former[2]	45.9	17.2	40.1	37.9	-	-
SEEM[25]	49.2	14.7	39.3	34.2	62.1	*
UniVS (Ours)	52.7	21.7	35.4	49.2	67.4	54.9

Table 3. Quantitative performance comparison among different unified segmentation models. '-' means that the model is inapplicable to this task and '*' means that the result is not reported. For the VIS task, the results are evaluated on the development set (1/10 of the training set, excluded during training).


a1) SepSA

a2) SepSA-entity

a3) SepSA-learnable

(a) Separated self-attention types

Self-attn Type	PVOS				
	$G^{th\&sf}$	G^{th}_{seen}	G^{th}_{unseen}	G^{sf}_{seen}	G^{sf}_{unseen}
a1	61.8	59.7	57.1	68.2	62.1
a2	57.9	58.4	55.0	63.5	54.5
a3	48.5	41.0	46.5	51.0	55.5

(b) Quantitative performance comparison on PVOS task

Self-attn Type	Used queries	RefYTVOS		
		J&F	J	F
a1	Prompt	38.7	36.1	41.3
a2	Prompt	55.7	53.9	57.5
a2	Prompt + Learnable	55.1	53.5	56.8

(c) Quantitative performance comparison on RefVOS task

Table 4. Ablation study on RefVOS tasks, where 'SepSA-e' and 'SepSA' mean that the separate self-attention mask is executed for each expression and all expressions, respectively.

entities. Lastly, 'SepSA-learnable' builds upon 'SepSA-entity' by allowing each prompt query to see all learnable queries to extract the overall image information.

To evaluate the impact of these three approaches on visual prompt-guided video segmentation tasks, we conducted an ablation study on the PVOS task, which involves simultaneous thing and stuff object segmentation. As shown in Table 4(b), the experimental results demonstrate that 'SepSA' performs the best, as it avoids content overflow between prompt and learnable queries.

Efficient Inference on Prompt-guided Segmentation. As shown in Table 4(c), UniVS can simultaneously process multiple prompt-guided targets in the RefVOS task by applying entity-wise separated self-attn mask (termed as SepSA-entity). This inference process is more efficient than the existing methods that often segment targets one by one.

Video Tasks	VIS		VPS	
Interval frames	YT21	OVIS	VIPSeg	
	mAP	mAP	VPQ	STQ
1	54.8	24.2	38.3	46.2
3	54.6	23.7	38.6	45.8
5	54.6	23.4	38.4	45.8
7	53.0	22.1	38.2	45.2
9	52.6	22.0	37.7	45.1

Table 5. Ablation study on the number of interval frames to detect newly appeared objects. The input clips include 5 frames.

Task	VIS	VSS	VPS	VOS	RefVOS	PVOS
Dataset	YT21	VSPW	VIPSeg	YT18	RefYT	VIPOSeg
FPS	20.2	15.3	10.4	17.5	20.0	11.9

Table 6. Inference speed of UniVS with ResNet50 backbone on a single V100 GPU.

Additionally, using only prompt queries can achieve higher performance than using both prompt and learnable queries.

The Detection of Newly Appeared Objects. For category-specified VS tasks, we investigate the impact of using different interval frames on detecting newly appeared objects. Since the VSS task only requires pixel-level category prediction without the need of instance-level tracking, it does not detect new objects. Therefore, we conduct ablation on the VIS and VPS tasks. The results are shown in Table 5. It can be observed that when the number of interval frames is smaller than the number of frames (*i.e.*, 5) in the input video clip, the performance is basically unaffected. However, if the interval frames exceed the number of frames in the input video clip, the performance is decreased by 1∼2%. This decline can be attributed to the missing of some newly appeared objects.

Inference speed. Table 6 shows the inference speed of UniVS with 640p video as input. Videos in YT21, YT18 and RefYT contain 1 ∼ 3 objects, whereas videos in VSPW, VIPSeg and VIPOSeg have more than 15 entities, whose inference speed is slower.

C.3. Generalization Ability

To further verify the generalization capability of UniVS, we try to train UniVS solely on the category-guided VS datasets but test on the prompt-guided VS datasets. In Table 7, we train UniVS only on category-specific VS tasks, including COCO, LVIS, YT21, OVIS and VIPSeg datasets. The testing is conducted on two prompt-guided VS tasks: VOS and PVOS. Experimental results demonstrate that UniVS exhibits comparable or even better performance on VOS and PVOS tasks, indicating its remarkable generalization ability. Additionally, we speculate that the significant performance improvement on DAVIS is due to its similarity to the training data distribution, while the slight performance drop on VIPOSeg is attributed to its inclusion of more diverse video scenes and objects, which exceeds the distribution of training data.

Training data		VOS	PVOS				
		DAVIS	VIPOSeg				
Category	Prompt	G^{th}	G^{th}_{seen}	G^{th}_{unsn}	G^{sf}_{seen}	G^{sf}_{unsn}	$G^{th\& sf}$
✓	✓	70.8	63.4	61.9	73.9	68.4	66.8
✓	✗	75.0	59.2	54.2	67.9	78.1	64.9

Table 7. Generalization ability of UniVS trained on category-guided VS tasks but tested on prompt-guided VS tasks. UniVS adopts SwinB backbone and trained on stages 1&2.

D. Visualization

VIS/VSS/VPS/VOS. Figs. 2 and 3 display the segmentation results predicted by our UniVS on VIS/VSS/VPS/VOS tasks. To enhance visualization, we use the same video for different VS tasks. Specifically, the thing categories for the VIS task is sourced from the OVIS dataset, while the thing and stuff categories for the VSS and VPS tasks are derived from the VIPSeg dataset. As for the VOS task, the visual prompts are obtained from the MOSE dataset. It can be observed that UniVS achieves satisfactory segmentation results across these tasks, demonstrating its excellent generalization capability.

RefVOS. Fig. 4 exhibits the video segmentation results with text expressions as prompts. We observe that UniVS can accurately segment objects in the video based on the given text prompts. This demonstrates that UniVS can effectively integrate language and video information, enabling cross-modal consistent segmentation.

PVOS. Fig. 5 displays the segmentation results for the PVOS task. The second and third rows compare the ground truth masks with the UniVS predicted masks, confirming the superiority of UniVS in visual prompt-guided thing and stuff entity segmentation. Additionally, it is worth noting that due to the high cost of video segmentation annotation, this dataset adopts a semi-automatic annotation approach, combining manual and algorithmic annotations. This may result in potential omissions or inaccuracies in the provided ground truth masks, such as the areas highlighted by the white bounding boxes. Therefore, UniVS also holds potential as a complementary method for dataset annotation in future endeavors.

In summary, UniVS demonstrates excellent general segmentation capability and can handle various VS tasks. It is not only suitable for category-guided segmentation but also performs well in almost all visual prompt-guided thing and stuff entity segmentation tasks. Meanwhile, UniVS showcases its ability in expression-guided cross-modal object segmentation tasks. Its multi-modal fusion capability and consistent segmentation performance make UniVS highly promising for integrating language and video information.

Video Demo. We provide more visualizations of the segmentation results on the project page. Please access the related content by clicking on the link `https://sites.google.com/view/unified-video-seg-univs`.

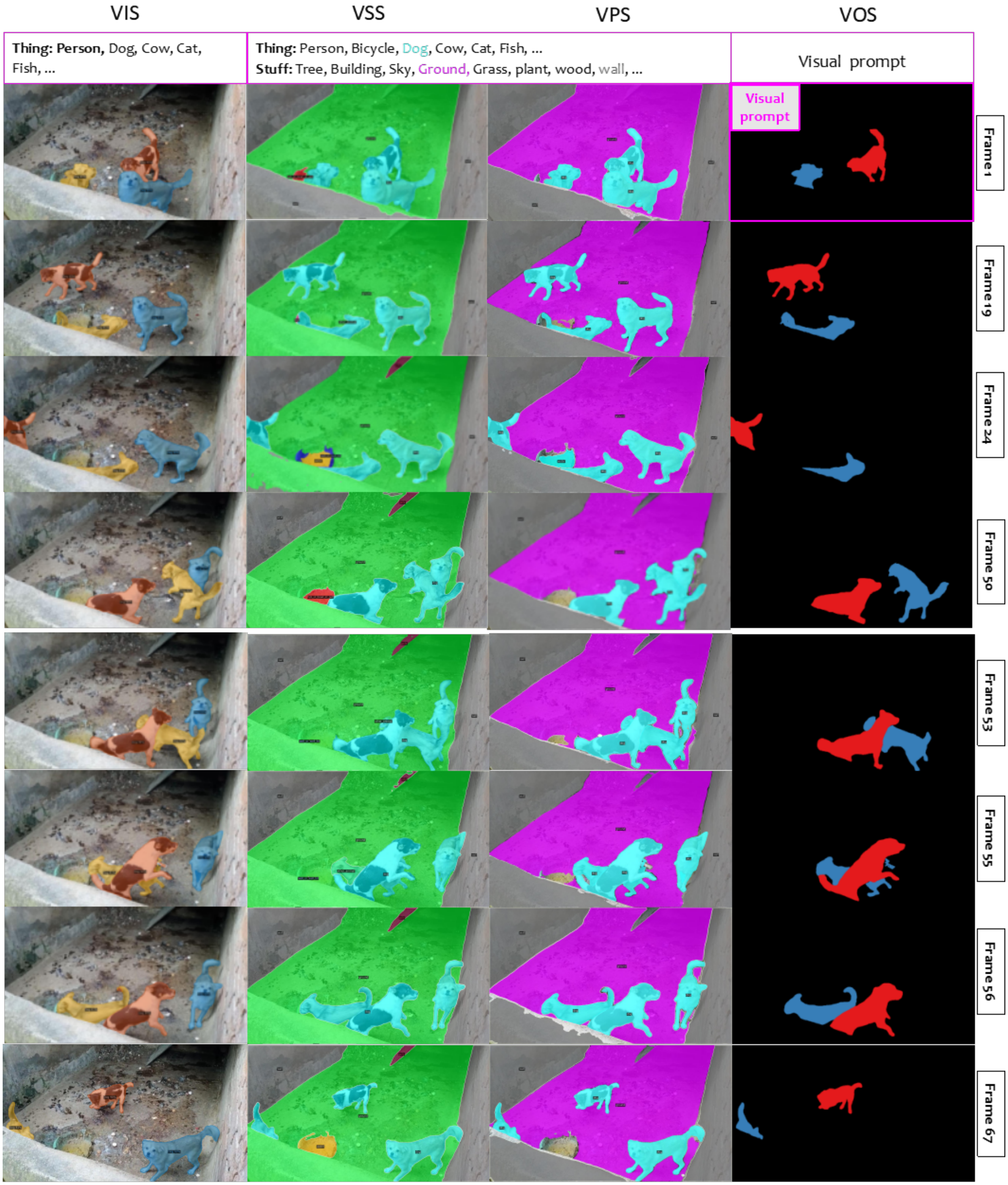

VIS
VSS
VPS
VOS
Thing: Person, Dog, Cow, Cat, Fish, ...
Thing: Person, Bicycle, Dog, Cow, Cat, Fish, ...
Stuff: Tree, Building, Sky, Ground, Grass, plant, wood, wall, ...
Visual prompt
Visual prompt
Frame 1
Frame 19
Frame 24
Frame 50
Frame 53
Frame 55
Frame 56
Frame 67

Figure 2. Visualization examples of UniVS on **VIS/VSS/VPS/VOS** tasks. The original videos come from the validation set of OVIS dataset, while the entity categories of VIS and VSS/VPS are from OVIS and VIPSeg datasets, respectively. The visual prompts are from the MOSE dataset.

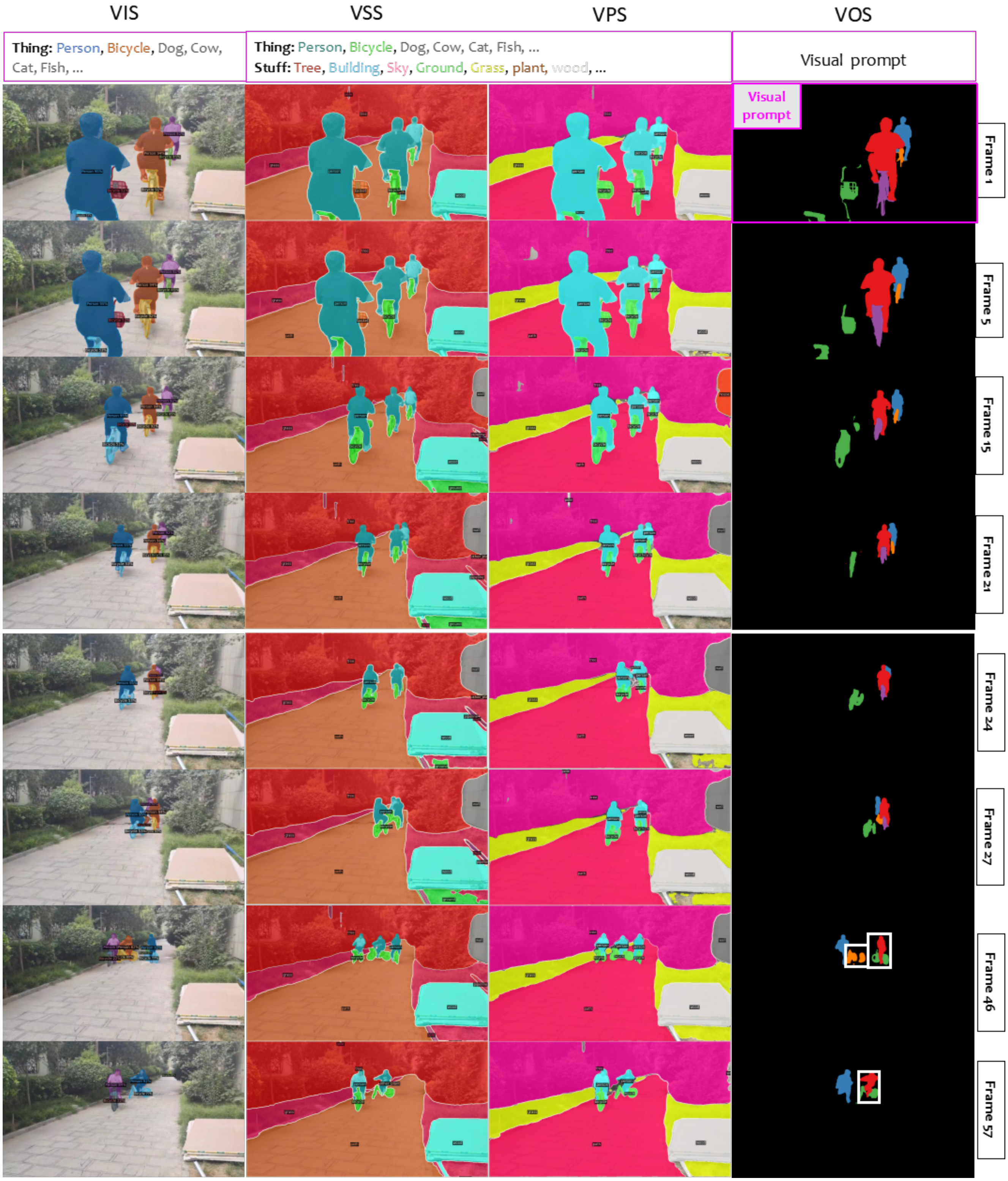

VIS
VSS
VPS
VOS
Thing: Person, Bicycle, Dog, Cow, Cat, Fish, ...
Thing: Person, Bicycle, Dog, Cow, Cat, Fish, ...
Stuff: Tree, Building, Sky, Ground, Grass, plant, wood, ...
Visual prompt
Visual prompt
Frame 1
Frame 5
Frame 15
Frame 21
Frame 24
Frame 27
Frame 46
Frame 57

Figure 3. Visualization examples of UniVS on **VIS/VSS/VPS/VOS** tasks. The original videos come from the validation set of OVIS dataset, while the entity categories of VIS and VSS/VPS are from OVIS and VIPSeg datasets, respectively. The visual prompts are from the MOSE dataset. For VOS task, we mark the incorrectly tracked objects in the last column with white bounding boxes.

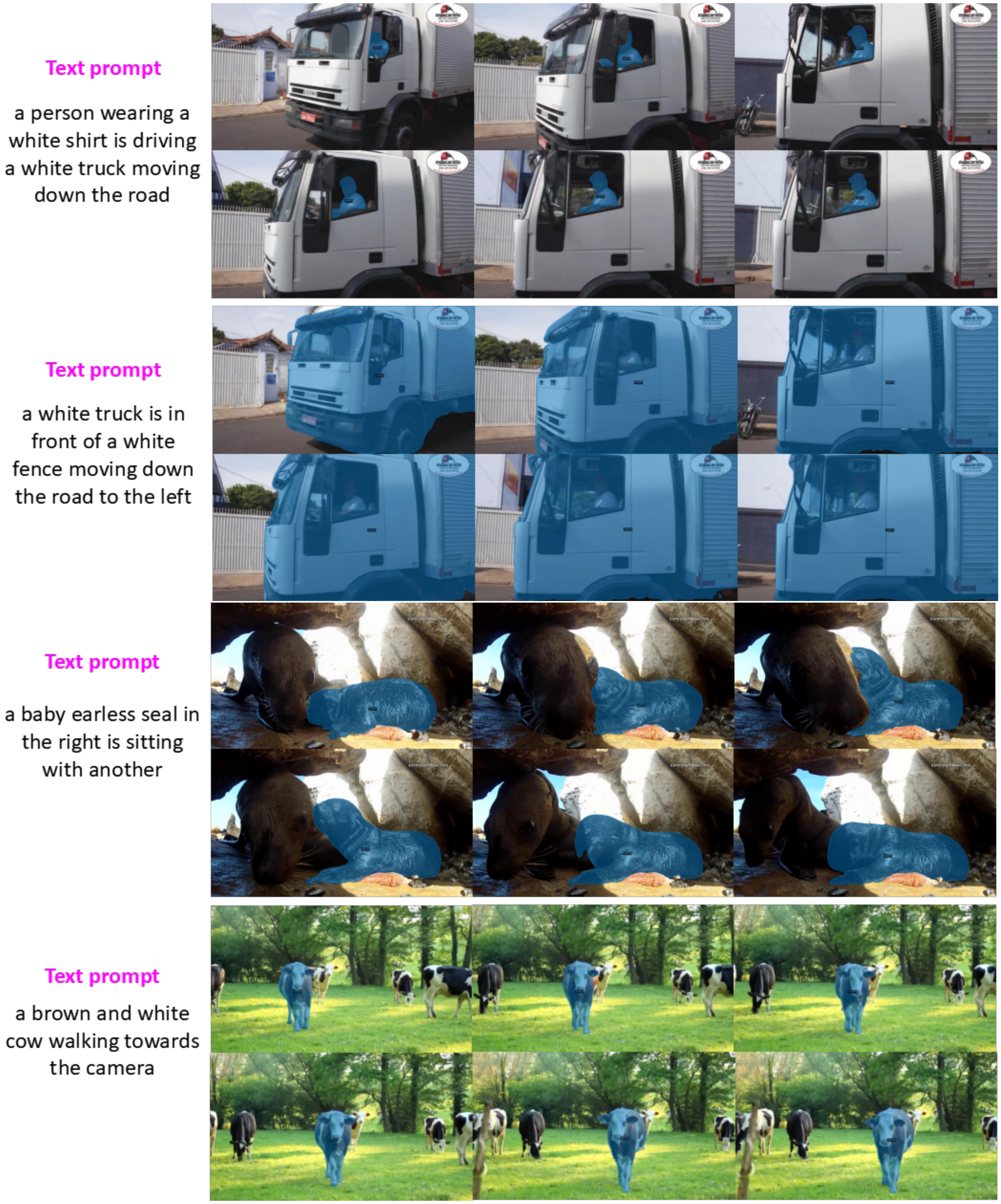

Text prompt
a person wearing a white shirt is driving a white truck moving down the road
Text prompt
a white truck is in front of a white fence moving down the road to the left
Text prompt
a baby earless seal in the right is sitting with another
Text prompt
a brown and white cow walking towards the camera

Figure 4. Visualization examples of UniVS with **text prompts** in the **RefVOS** task. The videos are from the RefYTVOS valid set, and the left side provides the expression per object.

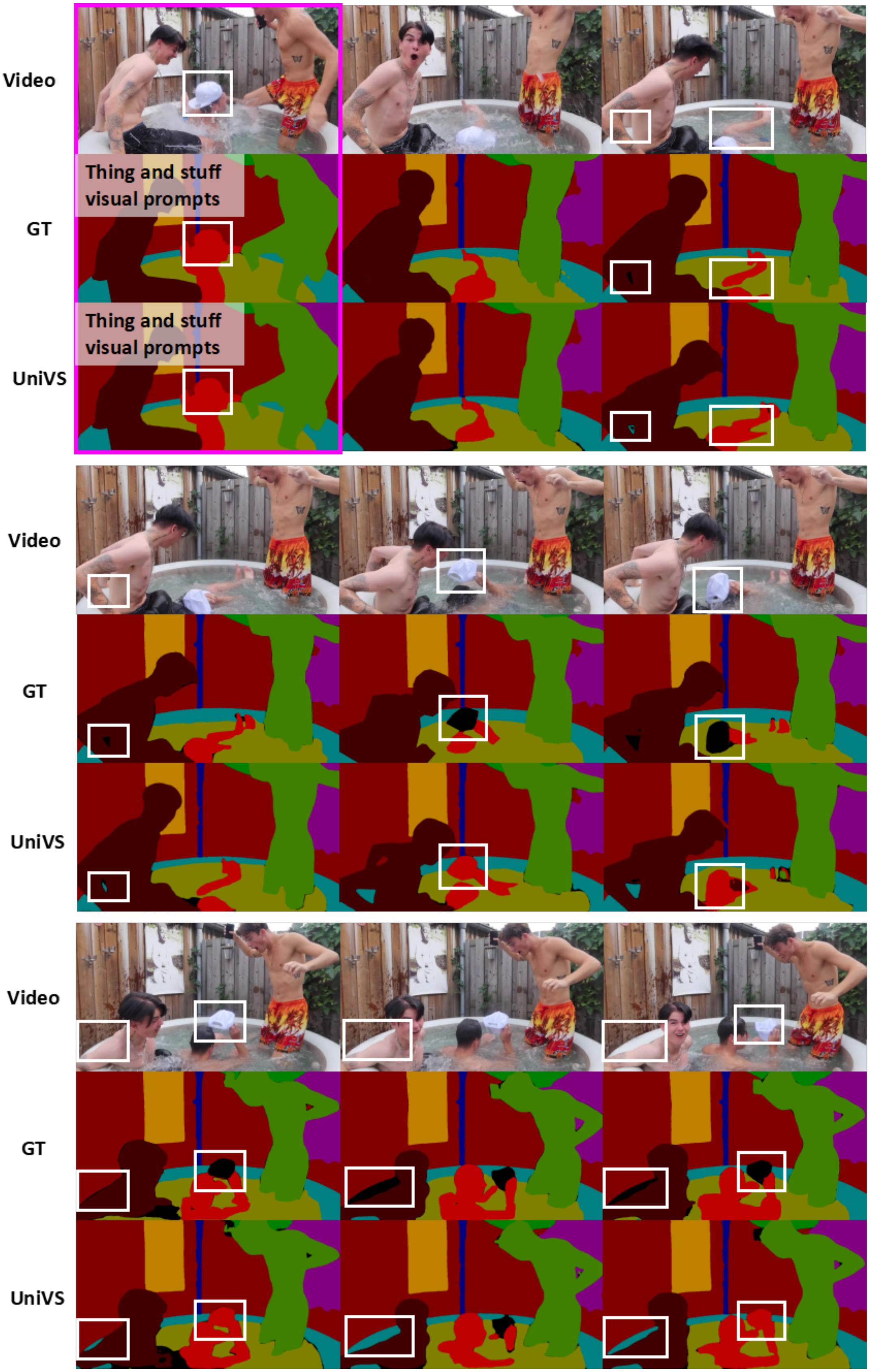

Video
GT
UniVS
Thing and stuff visual prompts
Thing and stuff visual prompts
Video
GT
UniVS
Video
GT
UniVS

Figure 5. Visualization examples of UniVS with **visual prompts** in the **PVOS** task. The video frames are from the VIPOSeg valid set, with the second row showing the ground truth masks and the last row displaying the predicted masks by our UniVS. Note that the visual prompts include both thing and stuff classes.

References

[1] Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15334–15342, 2021. 2

[2] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 3, 4

[3] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. 2

[4] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip HS Torr, and Song Bai. Mose: A new dataset for video object segmentation in complex scenes. *arXiv preprint arXiv:2302.01872*, 2023. 1

[5] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2

[6] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014. 2

[7] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9859–9868, 2020. 1

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2, 3

[9] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*, 2023. 3

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Eur. Conf. Comput. Vis.*, pages 740–755. Springer, 2014. 2

[12] Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-scale video panoptic segmentation in the wild: A benchmark. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21033–21043, 2022. 1

[13] Jiaxu Miao, Yunchao Wei, Yu Wu, Chen Liang, Guangrui Li, and Yi Yang. Vspw: A large-scale dataset for video scene parsing in the wild. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4133–4143, 2021. 1

[14] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 164–173, 2021. 2

[15] 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 Conf. Comput. Vis. Pattern Recog.*, 2016. 1

[16] 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 Conf. Comput. Vis. Pattern Recog.*, 2016. 1

[17] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip Torr, and Song Bai. Occluded video instance segmentation: Dataset and challenge. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 1

[18] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *Eur. Conf. Comput. Vis.*, pages 208–223. Springer, 2020. 1, 2

[19] Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, et al. Step: Segmenting and tracking every pixel. *arXiv preprint arXiv:2102.11859*, 2021. 1

[20] 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 *Eur. Conf. Comput. Vis.*, pages 585–601, 2018. 1, 2

[21] Yuanyou Xu, Zongxin Yang, and Yi Yang. Video object segmentation in panoptic wild scenes. *arXiv preprint arXiv:2305.04470*, 2023. 1

[22] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15325–15336, 2023. 3

[23] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Int. Conf. Comput. Vis.*, pages 5188–5197, 2019. 1

[24] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 3

[25] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *arXiv preprint arXiv:2304.06718*, 2023. 3, 4