Universal Instance Perception as Object Discovery and Retrieval
—— Supplementary Material ——

Bin Yan[1]*, Yi Jiang[2,†], Jiannan Wu[3], Dong Wang[1],
Ping Luo[3], Zehuan Yuan[2], Huchuan Lu[1,4,†]
[1] School of Information and Communication Engineering, Dalian University of Technology, China
[2] ByteDance [3] The University of Hong Kong [4] Peng Cheng Laboratory

1. Appendix

In this appendix, we present more details about the training process and loss functions in 1.1 and 1.2, network architecture in 1.3, as well as more analysis and visualizations for better understanding in 1.4.

1.1. Loss Functions

We present detailed loss functions for better readability. First, $\mathcal{L}_{\text{retrieve}}$ and $\mathcal{L}_{\text{box}}$ are used across all three stages. Second, to learn mask representations from coarse boxes [16] and fine mask annotations [9, 15, 21, 23, 25], UNINEXT uses $\mathcal{L}_{\text{mask}}^{\text{boxinst}}$ in the first stage and $\mathcal{L}_{\text{mask}}$ in the next two stages respectively. Finally, to associate instances on different frames [13, 23, 24], UNINEXT additionally adopts $\mathcal{L}_{\text{embed}}$ in the last stage.

$\underline{\boldsymbol{\mathcal{L}}_{\textbf{retrieve}}}$. Given the raw instance-prompt matching score s, the normalized matching probability p is computed as $p = \sigma(s)$, where σ is sigmoid function. Then $\mathcal{L}_{\text{retrieve}}$ can be written as the form of Focal loss [8].

$$\mathcal{L}_{\text{retrieve}}(p_{\text{t}}) = -\alpha_{\text{t}}(1-p_{\text{t}})^{\gamma}\log(p_{\text{t}}). \tag{1}$$

$$p_{\text{t}} = \begin{cases} p & \text{if matched} \\ 1-p & \text{otherwise.} \end{cases} \tag{2}$$

γ and α are 2 and 0.25 respectively.

$\underline{\boldsymbol{\mathcal{L}}_{\textbf{box}}}$. Following DETR-like methods [2, 29], $\mathcal{L}_{\text{box}}$ consists of two terms, GIoU Loss [14] and ℓ_1 loss:

$$\mathcal{L}_{\text{box}}(b,\hat{b}) = \lambda_{giou}\mathcal{L}_{\text{giou}}(b,\hat{b}) + \lambda_{L_1}\|b-\hat{b}\|. \tag{3}$$

$$\mathcal{L}_{\text{giou}}(b,\hat{b}) = 1 - IoU(b,\hat{b}) + \frac{A^c(b,\hat{b}) - U(b,\hat{b})}{A^c(b,\hat{b})}, \tag{4}$$

where $A^c(b,\hat{b})$ is the area of the smallest box containing b and $\hat{b}$. $U(b,\hat{b})$ is the area of the union of b and $\hat{b}$.

$\underline{\boldsymbol{\mathcal{L}}_{\textbf{mask}}}$. For datasets with mask annotations [9, 15, 21, 23, 25], Focal Loss [8] and Dice Loss [10] are adopted.

$$\mathcal{L}_{\text{mask}}(m,\hat{m}) = \lambda_{focal}\mathcal{L}_{\text{focal}}(m,\hat{m}) + \lambda_{dice}\mathcal{L}_{\text{dice}}(m,\hat{m}). \tag{5}$$

$$\mathcal{L}_{\text{dice}}(m,\hat{m}) = 1 - \frac{2m\hat{m}+1}{\hat{m}+m+1}, \tag{6}$$

where m and $\hat{m}$ are binary GT masks and predicted masks after sigmoid activation respectively.

$\underline{\boldsymbol{\mathcal{L}}_{\textbf{mask}}^{\textbf{boxinst}}}$. For Objects365 [16] without mask annotations, UNINEXT uses Projection Loss and Pairwise Affinity Loss like BoxInst [18], which can learn mask prediction only based on box-level annotations.

$$\mathcal{L}_{\text{mask}}^{\text{boxinst}}(b,\hat{m}) = \mathcal{L}_{\text{proj}}(b,\hat{m}) + \mathcal{L}_{\text{pairwise}}(b,\hat{m}). \tag{7}$$

$$\begin{aligned}\mathcal{L}_{\text{proj}}(b,\hat{m}) = &\mathcal{L}_{\text{dice}}(\text{proj}_{\text{x}}(b), \text{proj}_{\text{x}}(\hat{m})) + \\ &\mathcal{L}_{\text{dice}}(\text{proj}_{\text{y}}(b), \text{proj}_{\text{y}}(\hat{m})).\end{aligned} \tag{8}$$

$$\mathcal{L}_{\text{pairwise}} = -\frac{1}{N}\sum_{e\in E_{in}} \mathbb{1}_{\{S_e \geq \tau\}} \log P(y_e=1). \tag{9}$$

$$P(y_e=1) = \hat{m}_{i,j}\cdot\hat{m}_{k,l} + (1-\hat{m}_{i,j})\cdot(1-\hat{m}_{k,l}). \tag{10}$$

$$S_e = S(c_{i,j}, c_{l,k}) = \exp\left(-\frac{||c_{i,j}-c_{l,k}||}{\theta}\right), \tag{11}$$

where $y_e = 1$ means the two pixels have the same ground-truth label. S_e is the color similarity of the edge e. $c_{i,j}$ and $c_{l,k}$ are respectively the LAB color vectors of the two pixels (i,j) and (l,k) linked by the edge. θ is 2 in this work.

$\underline{\boldsymbol{\mathcal{L}}_{\textbf{embed}}}$. UNINEXT uses contrastive loss [20] to train discriminative embeddings for associating instances on different frames.

$$\mathcal{L}_{\text{embed}} = \log[1 + \sum_{\mathbf{k}^+}\sum_{\mathbf{k}^-}\exp(\mathbf{v}\cdot\mathbf{k}^- - \mathbf{v}\cdot\mathbf{k}^+)], \tag{12}$$

where k^+ and k^- are positive and negative feature embeddings from the reference frame. For each instance in the key frame, v is the feature embedding with the lowest cost.

*This work was performed while Bin Yan worked as an intern at ByteDance. Email: yan_bin@mail.dlut.edu.cn. † Corresponding authors: jiangyi.enjoy@bytedance.com, lhchuan@dlut.edu.cn.

Table 1. Details in training. Step is the time to reduce the learning rate.

Stage	Task	Dataset	Sampling Weight	Batch Size	Short	Long	Num GPU	Lr	Max Iter	Step
I	OD&IS	Objects365 [16]	1	2	480 ~ 800	1333	32	0.0002	340741	312346
II	OD&IS	COCO [9]	1	2	480 ~ 800	1333	16	0.0002	91990	76658
	REC&RES	RefCOCO/g/+ [12,25]	1	2	480 ~ 800	1333				
III	SOT&VOS	LaSOT [4]	0.20	2	480 ~ 800	1333	16	0.0001	180000	150000
		GOT10K [6]	0.20	2	480 ~ 800	1333				
		TrackingNet [11]	0.20	2	480 ~ 800	1333				
		Youtube-VOS [21]	0.20	2	320 ~ 640	768				
		COCO [9]	0.20	2	480 ~ 800	1333				
	MOT&MOTS	BDD-obj-det [24]	0.18	2	480 ~ 800	1333				
		BDD-box-track [24]	0.72	2	480 ~ 800	1333				
		BDD-inst-seg [24]	0.02	2	480 ~ 800	1333				
		BDD-seg-track [24]	0.08	2	480 ~ 800	1333				
	VIS	Youtube-VIS-19 [23]	0.34	4	320 ~ 640	768				
		OVIS [13]	0.17	2	480 ~ 800	1333				
		COCO [9]	0.51	2	480 ~ 800	1333				
	R-VOS	Ref-Youtube-VOS [15]	0.33	2	320 ~ 640	768				
		RefCOCO/g/+ [12,25]	0.67	2	480 ~ 800	1333				

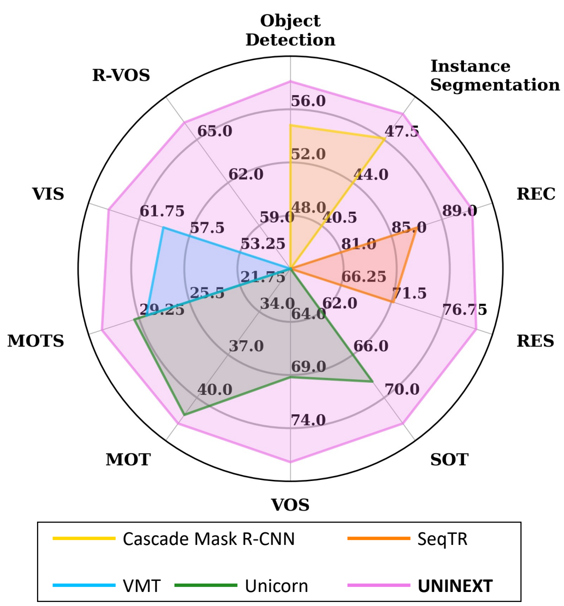

Figure 1. Better view in color on screen.

1.2. Training Process

The detailed hyperparameters during training are shown in Tab 1. The whole training process consists of three stages. In each stage, the `StepLR` learning rate scheduler is adopted. The learning rate drops by a factor of 10 after the given steps. For multi-dataset training, we follow the implementation of Detic [27], which randomly samples data from different tasks and then computes them on different GPUs in one iteration. Besides, the multi-scale training technique is used across all datasets in all stages. Take the pre-training on Objects365 [16] as an example, the original images are resized such that the shortest side is at least 480 and at most 800 pixels while the longest side is at most 1333. We use this as the default setting except on Youtube-VOS [21], Youtube-VIS-2019 [23], and Ref-Youtube-VOS [15]. A lower resolution with the shortest side ranging from 320 to 640 and the longest side not exceeding 768 is applied to these datasets [15, 21, 23], following previous works [3, 19, 20].

Specifically, in the first stage, the model is pretrained on Objects365 [16] for about 340K iterations (12 epochs) and the learning rate drops on the 11th epoch. In the second stage, we finetune UNINEXT on COCO [9] and RefCOCO/g/+ [12, 25] jointly for 12 epochs. In the third stage, UNINEXT is further finetuned for diverse video-level tasks. To guarantee balanced performance on various benchmarks, we set the data sampling ratios as (SOT&VOS):(MOT&MOTS):VIS:R-VOS = 1:1:1:1. For each task, 45K iterations are allocated, thus bringing 180K iterations in total for the third stage. Besides, to avoid forgetting previously learned knowledge on image-level tasks, we also generate pseudo videos from COCO [9] and RefCOCO/g/+ [12, 25] and mix them with training data of VIS [13, 23] and R-VOS [15] respectively.

1.3. Network Architecture

To transform the enhanced visual features F'_v and prompt features F'_p into the final instance predictions, an encoder-decoder Transformer architecture is adopted. Based on the original architecture in two-stage Deformable DETR [29], UNINEXT makes the following improvements:

- **Introducing a mask head for segmentation.** To predict high-quality masks, UNINEXT introduces a mask head [17] based on dynamic convolutions. Specifically, first an MLP is used to transform instance em-

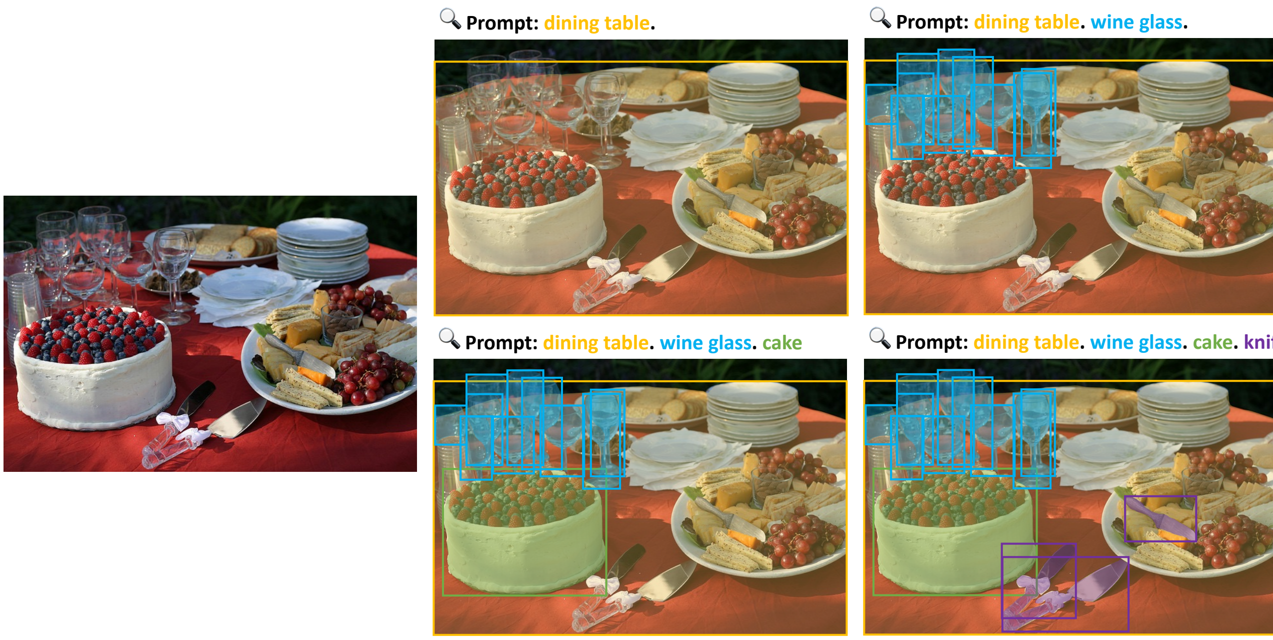

Figure 2. Illustration of **retrieval by category names**. UNINEXT can flexibly perceive objects of different categories by changing the input prompts. Better view in color on screen.

beddings into a group of parameters ω. Then these parameters are used to perform three-layer 1×1 convolutions with feature maps, obtaining masks of instances.

- **Replacing one-to-one Hungarian matching with one-to-many SimOTA [5].** Traditional Hungarian matching forces one GT to be only assigned to one query, leaving most of the queries negative. UNINEXT uses SimOTA [5], which enables multiple queries to be matched with one GT. This strategy can provide more positive samples and speed up convergence. During inference, UNINEXT uses NMS to remove duplicated predictions.
- **Adding an IoU branch.** UNINEXT adds an IoU branch to reflect the quality of the predicted boxes. During training, IoU does not affect the label assignment. During inference, the final scores are the geometric mean of the instance-prompt matching scores (after sigmoid) and the IoU scores.
- **Adding some techniques in DINO [26].** To further improve the performance, UNINEXT introduces some techniques [26], including contrastive DN, mixed query selection, and look forward twice.

1.4. Analysis and Visualizations

Analysis. We compare UNINEXT with other competitive counterparts, which can handle multiple instance-level perception tasks. The opponents include Cascade Mask R-CNN [1] for object detection and instance segmentation, SeqTR [28] for REC and RES, VMT [7] for MOTS and VIS, and Unicorn [22] for SOT, VOS, MOT, and MOTS. As shown in Figure 1, UNINEXT outperforms them and achieve state-of-the-art performance on all 10 tasks.

Retrieval by Category Names. As shown in Figure 2, UNINEXT can flexibly detect and segment objects of different categories by taking the corresponding category names as the prompts. For example, when taking "dining table. wine glass. cake. knife" as the prompts, UNINEXT would only perceive dining tables, wine glasses, cakes, and knives. Furthermore, benefiting from the flexible retrieval formulation, UNINEXT also has the potential for zero-shot (open-vocabulary) object detection. However, open-vocabulary object detection is beyond the scope of our paper and we leave it for future works.

Retrieval by Language Expressions. We provide some visualizations for retrieval by language expressions in Figure 3. UNINEXT can accurately locate the target referred by the given language expression when there are many similar distractors. This demonstrates that our method can not only perceive objects but also understand their relationships in positions (left, middle, right, etc) and sizes (taller, etc).

Retrieval by Target Annotations. Our method supports annotations in both boxes (SOT) and masks (VOS) formats. Although there is only box-level annotation for SOT, we obtain the target prior by filling the region within the given box with 1 and leaving other regions as 0. As shown in Figure 4, UNINEXT can precisely track and segment the targets in complex scenarios, given the annotation in the first frame.

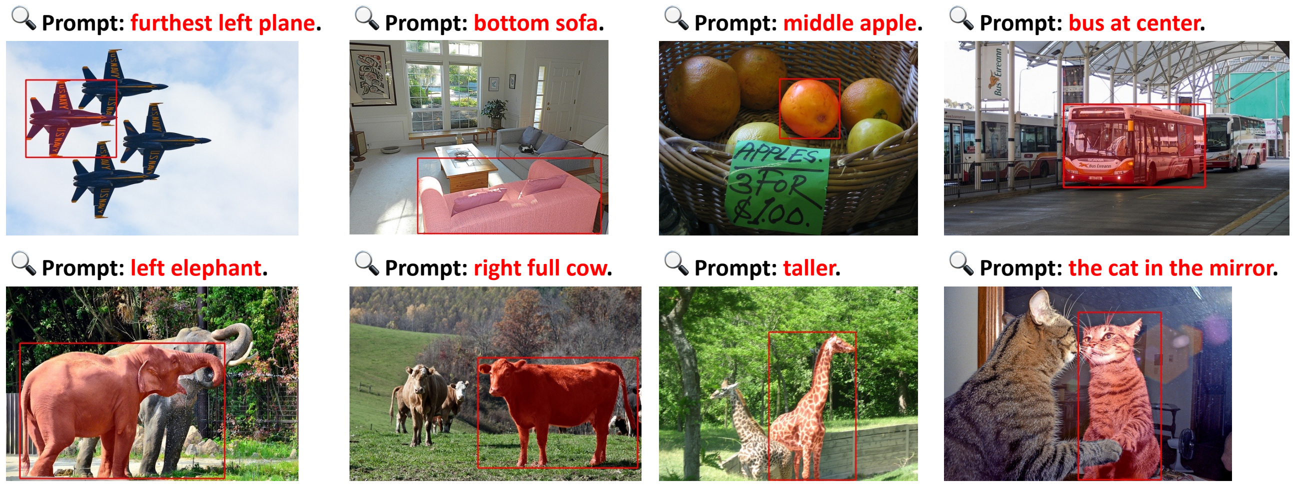

Figure 3. Illustration of **retrieval by language expressions**. Better view in color on screen.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: high quality object detection and instance segmentation. *TPAMI*, 2019. 3

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1

[3] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *NeurIPS*, 2021. 2

[4] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019. 2

[5] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3

[6] Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 2019. 2

[7] Lei Ke, Henghui Ding, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Video mask transfiner for high-quality video instance segmentation. *ECCV*, 2022. 3

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1

[9] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 2

[10] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016. 1

[11] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018. 2

[12] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016. 2

[13] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *IJCV*, 2022. 1, 2

[14] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019. 1

[15] Seonguk Seo, Joon-Young Lee, and Bohyung Han. URVOS: Unified referring video object segmentation network with a large-scale benchmark. In *ECCV*, 2020. 1, 2

[16] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019. 1, 2

[17] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020. 2

[18] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. BoxInst: High-performance instance segmentation with box annotations. In *CVPR*, 2021. 1

[19] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, 2022. 2

[20] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In defense of online models for video instance segmentation. *ECCV*, 2022. 1, 2

[21] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. YouTube-VOS: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 1, 2

[22] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *ECCV*, 2022. 3

[23] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 1, 2

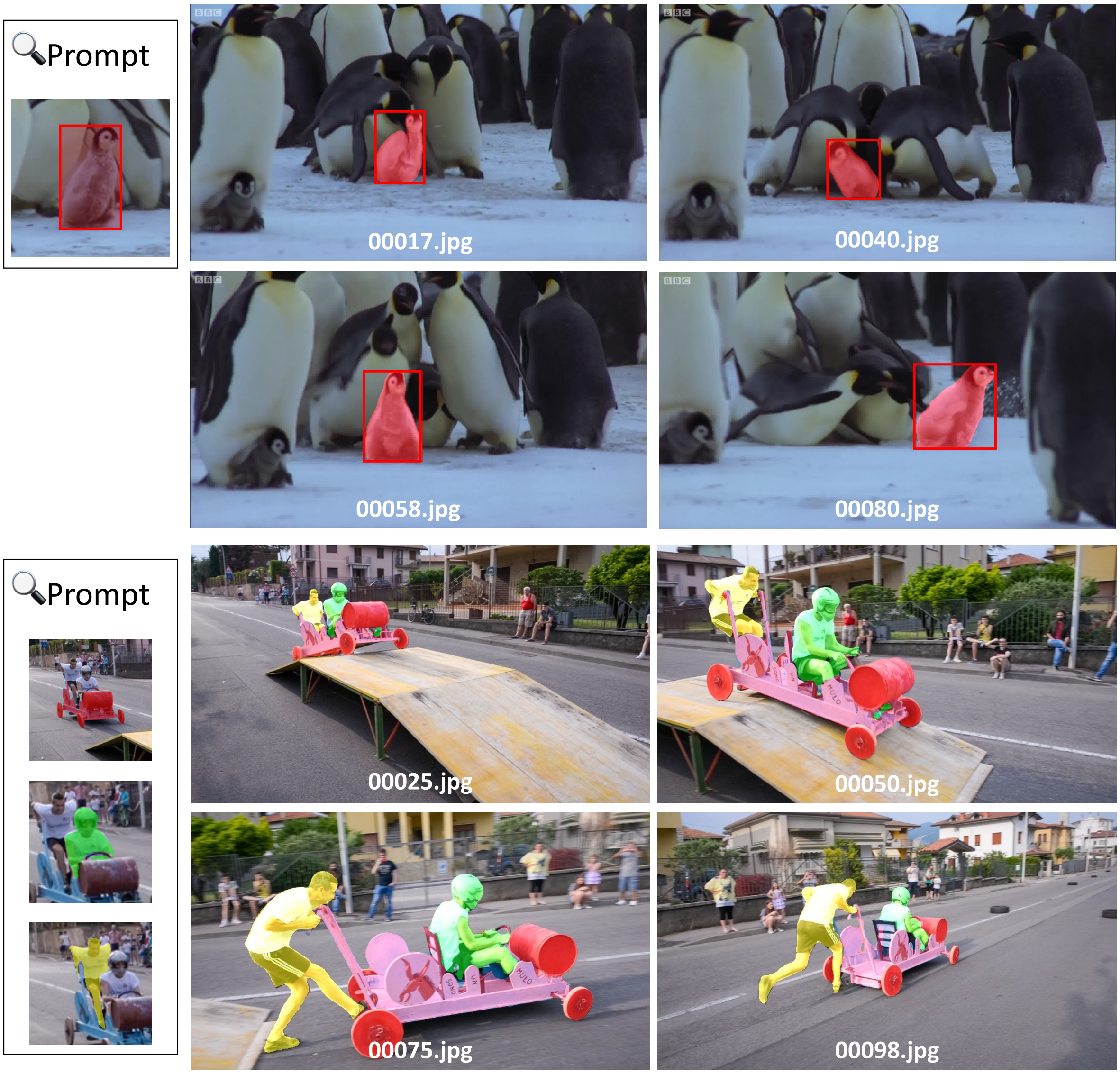

Figure 4. Illustration of **retrieval by target annotations**. UNINEXT can flexibly perceive different objects according to the box or mask annotations given in the first frame. Better view in color on screen.

[24] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020. 1, 2

[25] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016. 1, 2

[26] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. DINO: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 3

[27] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 2

[28] Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. SeqTR: A simple yet universal network for visual grounding. *ECCV*, 2022. 3

[29] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 1, 2